Ensembles of Text and Time-Series Models for Automatic Generation of Financial Trading Signals

By

Omar Abdul Bari

Submitted to the graduate degree program in Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Chair Arvin Agah

Joseph Evans

Andrew Gill

Jerzy Gryzmala-Busse

Sara Wilson

Date Defended: August 9, 2016

The Dissertation Committee for Omar Abdul Bari certifies that this is the approved version of the following dissertation:

Ensembles of Text and Time-Series Models for Automatic Generation of Financial Trading Signals

Chair Arvin Agah

Data Approved:

Abstract

Event Studies in Finance have focused on traditional news headlines to assess the impact an event has on a traded company. The increased proliferation of news and information produced by social media content has disrupted this trend. Although researchers have begun to identify trading opportunities from social media platforms, such as Twitter, almost all techniques use a general sentiment from large collections of tweets. Though useful, general sentiment does not provide an opportunity to indicate specific events worthy of affecting stock prices.

This work presents an event clustering algorithm, utilizing natural language processing techniques to generate newsworthy events from Twitter, which have the potential to influence stock prices in the same manner as traditional news headlines. The event clustering method addresses the effects of pre-news and lagged-news, two peculiarities that appear when connecting trading and news, regardless of the medium. Pre-news signifies a finding where stock prices move in advance of a news release. Lagged-news refers to follow-up or late-arriving news, adding redundancy in making trading decisions.

For events generated by the proposed clustering algorithm, we have designed and implemented novel language and time-series techniques – incorporating Event Studies and Machine Learning to produce an actionable system that can guide trading decisions. Of the various methods considered, the emphasis was particularly on the state-of-the-art established methods versus modern Deep Learning techniques. The recommended prediction algorithms provide investing strategies with profitable risk-adjusted returns.

The suggested language models present Annualized Sharpe Ratios (risk-adjusted returns) in the 5 to 11 range, while time-series models produce in the 2 to 3 range (without transaction costs).

A close investigation of the distribution of returns confirms the encouraging Sharpe Ratios by identifying most outliers as significant positive gains. Additionally, Machine Learning metrics of precision, recall, and accuracy are discussed alongside financial metrics in hopes of bridging the gap between academia and industry in the field of Computational Finance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Summary

Event Studies research focuses on the statistical impact that an event has on a traded company [MacKinlay 1997]. In Finance, a financial press-release announcing company earnings is an example of an event. Unlike earnings announcements, media events may arise unexpectedly. By using the framework of an Event Study, this dissertation explores unexpected events in modern media – particularly Twitter [Twitter 2016]. Measuring statistical impact is not the central goal. Instead, listed here are the selected implementation objectives. By utilizing natural language processing, the goal is to identify events on Twitter that influence stock prices of firms. Text and time-series models are generated by applying machine learning techniques in order to classify events. Quantitative trading decisions are developed by associating prediction outputs as trading signals. The implementation objectives combine Event Studies and Machine Learning to produce an actionable system that can guide trading decisions.

1.2 Motivation

With advancements in computation power, Market investors adapt by relying heavily on empirical findings as opposed to classical economic theories. The fields of behavioral and mathematical finance emphasize a shift towards interpreting empirical findings and market realities. The conviction of using observations for asset price predictions originates from the studies of data science and mathematical finance. Any edge in prediction power via empirical evidence is considered essential to the investors and financial agencies.

Specifically, of interest to this work is the way institutions and traders absorb information from expanding amounts of unstructured financial data. The foremost concern is to understand the economic impact caused by unexpected news. Technological innovations have changed the way that news disseminates. Therefore, an emphasis is placed in this work on modern news mediums. Case in point of this transformation is Twitter, a popular social media platform, providing a means for posting unverified bits of information. To financial markets, any medium that breaks news first is relevant. Moreover, tweets have the power to move markets. A hack on the Twitter account of the Associated Press (AP) on April 23, 2013, stresses this power. The Dow Jones Industrial Average dropped 128 points seconds after a tweet from the hacked AP account contained the following line "*Breaking: Two Explosions in the White House and Barack Obama is injured.*".

The focus of this work is to use predictive analytics to gain insights on reactions to financial events. As a contribution, we address procedures to reduce the challenges of modern media. We study ensembles of novel language and time-series models to enhance

the current state of Event Studies research.

1.3 Dissertation Structure

This dissertation is structured into six chapters. Chapter 2 highlights the related work and the necessary background. Chapter 3 presents the research foundation with hypotheses, design, and methodology. Chapter 4 provides more details into the system design. Chapter 5 highlights the key insights and results obtained by using the presented methodology and design. Chapter 6 concludes with a discussion of hypotheses, contributions, limitations, and potentials for future work.

Chapter 2

Background and Related Work

2.1 Overview

Event Studies in Finance investigate the impact that events have on the value of traded firms. The impact is quantified by the change in stock prices immediately following an event. Learning the measure of how events effect prices are of substantial worth. The large sums of money spent by the financial industry in revealing these statistical patterns are evidence of its significance. The investigation, however, is vast. The types of events are numerous. Hence, we follow a careful procedure to properly implement an Event Study. A principal study from economics provides the guideline on how best to structure an Event Study [MacKinlay 1997].

Event Analysis research from Computer Science often starts by considering the efficient-market hypothesis [Fama 1965]. This introduces the idea that financial markets are efficient. Efficient markets add to the view that stock prices follow a "random walk", meaning that new information is reflected instantly in the price of the stock. Given this understanding, there are no opportunities to predict the movement of a stock. While this does sound counter-intuitive, the hypothesis has had its critics. Thus, once done stating the efficient-market hypothesis, it is common for authors to provide empirical evidence showing that predictability of prices is, in fact, possible. Even though there are numerous papers that would both support and conflict the efficient-market

hypothesis [Malkiel 2003], it is more productive alternatively to see where Computer Science techniques and tools can help in furthering Computational Finance literature of event analysis. Accordingly, we believe evidence confirming the ability to predict prices appears pertinent to the emerging disciplines of behavioral finance and investor sentiments.

Behavioral finance and theories on investor sentiments both heed to the general idea that perceptions influence market outcomes. Perceptions guide investment decisions. Therefore, while not simple, predictability of prices may be plausible. The behavioral finance belief in perceptions explains why certain trading activities show signs of overreaction/underreaction [Daniel *et al.* 1998].

It is beneficial to know where within the financial literature price predictability fits. However, the main objective is to construct and execute the Event Analysis framework. The structure in turn produces the end goal: Making better trading decisions.

In order to complete a background on Event Analysis, we have to recognize both fundamental and technical analysis. Fundamental analysis considers how new information absorbs into markets. Fundamental information is of an intrinsic sort, one that has an effect on firm value. This information is represented by either unstructured or semi-structured data. Examples being SEC Filings, earnings, political/economic situations, and news. The second branch, technical analysis, studies movements and patterns in historical market data, usually focusing on a stock's price and trading volume.

Event analysis investigates how markets absorb new unstructured or semi-structured fundamental data. Technical analysis, while not required, is welcomed. At any given time,

the market price incorporates all perception and intrinsic value of a company. Therefore, technical analysis helps with the understanding of past information about the stock.

Specifically, this background will arrange the current situation of the Event Analysis framework. The ensuing list orders the examination:

- Event Analysis (Framework and Fundamental Analysis)
- Natural Language Processing (Word Representations and Sentiment)
- Time-Series Analysis (Data Representation, Technical Analysis, and Time-Series Regressors and Classifiers)

It should be noted that the financial information which is considered is news. Taken into account are both traditional and social media sources (e.g., Thomson Reuters, Dow Jones, Twitter, and Financial Times).

2.2 Event Analysis

2.2.1 Event Study

In order to measure the effect that breaking news has on prices, it is best to align the timestamp of each news story with its market data. The x-axis of the Event Horizon in Figure 2.1 marks the time components in an Event Study. The horizon consists of a fixed event time surrounded by the pre/post-event regions. To illustrate, the event time is set equal to the timestamp of a news release. A fixed time of 20 time units is selected to examine the study before and after that event. As such, the components are event time T, pre-event region $[T-20, T)$, and the post-event region $(T, T+20]$. Lastly, to

synchronize market data with the Event Horizon, it is imperative to set the cumulative stock return equal to 0% at time $T - 20$ minutes. Doing so captures the performance of prices throughout a 40 time unit event window. A comparable and fixed Event Horizon is now available to align all news releases. It is at this moment, easier to collect separately and average the price impact of all positive/negative news. Selecting the amount of time needed for pre/post-event analysis is a matter of choice. It is, however, highly dependent on whether the trading frequency is in seconds, minutes, days, months or years.

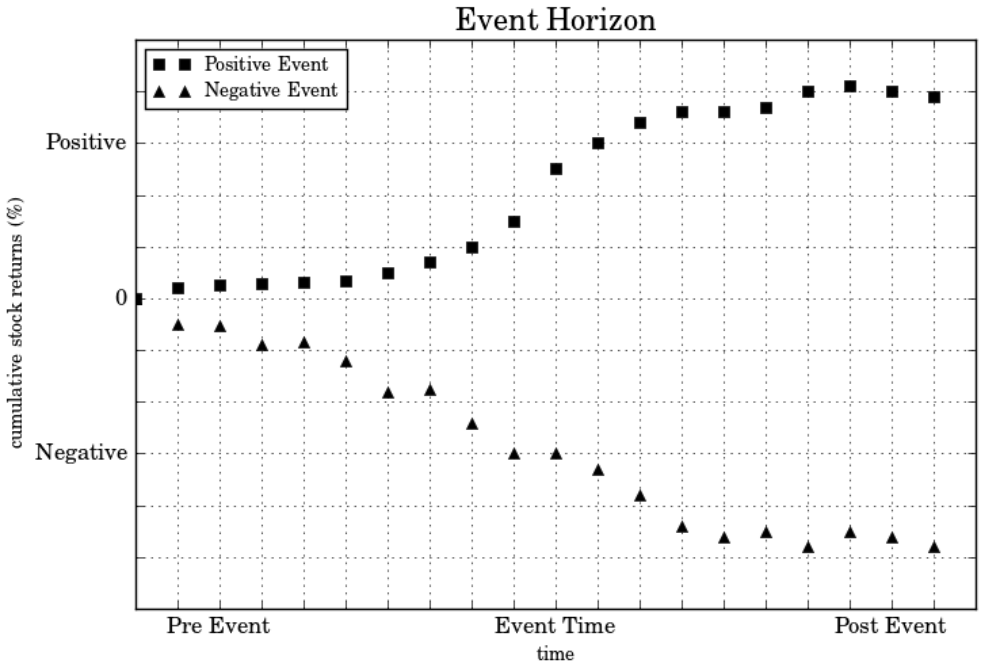


Figure 2.1: Event Horizon

Figure 2.1 also highlights key insights that appear in many published papers [GroB-KluBmann & Hautsch 2011, Leinweber & Sisk 2011, MacKinlay 1997, Tetlock *et al.* 2008]. First, cumulative returns show that investors respond well to good

stories and poorly to bad ones. The second finding, not as obvious, establishes some presence of a pre-news effect. Prior to a press announcement, stock prices prematurely move in the expected direction. Financial institutions and traders ingest a variety of sources similar to journalists. For this reason, traditional media has a systematic form of information leak. The pre-news effect is an essential concept discussed in detail later. Regardless of the pre-news effect, it is apparent that prices continue in their respective direction once the story is public. It is strengthening the view of behavioral finance where investor perception guides investment decisions.

Without the availability of a standardized dataset, researchers frequently select news from a set of traditional media outlets. Major sources studied are the Wall Street Journal and Dow Jones News [Chan 2003, Tetlock *et al.* 2008], Yahoo Finance [Schumaker & Chen 2009], Financial Times [Soni *et al.* 2007], Forbes [Rachlin *et al.* 2007], and Thomson Reuters News Analytics [Leinweber & Sisk 2011]. More recently, social media are deemed an important area of research. The most influential being Twitter [Bollen *et al.* 2011, Makrehchi *et al.* 2013, Ruiz *et al.* 2012, Sul *et al.* 2014].

Research is not restricted to only prices. It is shown relevant to test all segments of market data. For example, significant market responses to events are noticed in both volatility and cumulative trading volume [GroB-KluBmann & Hautsch 2011]. These two market segments indirectly influence average trade sizes and bid-ask spreads. By increasing price fluctuations, a rise in volatility heightens firm risk. Increased bid-ask spreads make it more difficult to buy and sell stocks by accruing larger transaction

costs. Thus, the impact that events have on different components of market data plays a big role in decision making for optimizing a trading strategy. That being said, in research, prices/returns are the most featured market data components. Findings show a notable contrast in returns between stocks with events and without events [Chan 2003]. A contrast such as this shows that events do indeed have an impact on the outcome of markets. It is an important result for advocates of Event Analysis.

Presented earlier is the economic justification for Event Studies, and the empirical evidence that supports it. Next, the role computation plays is considered. Even in weakly efficient markets, the time available to predict prices and to make profits is diminishing. The quicker that one can identify a market-moving article, the earlier one gets to complete a transaction. The news is unstructured. Therefore, natural language processing techniques are required. Machine learning plays a significant role because automated data-driven decisions are much faster than standard analysis. Technological innovations guiding how individuals connect to markets have shaped the academic influence of Computational Science in Finance. Many regard this field as Computational Finance.

The additional research that highlights relevant trends and techniques of Machine Learning for finance and event trading is listed here: [Antweiler & Frank 2004, Chung *et al.* 2012, Chatrath *et al.* 2014, Fasanghari & Montazer 2010, Ghiassi *et al.* 2013, Groth & Muntermann 2011, Hagenau *et al.* 2013, Huang *et al.* 2010, Kontopoulos *et al.* 2013, Liao *et al.* 2014, Mao *et al.* 2013, Mittermayer 2004, Nuij *et al.* 2014, Ruiz *et al.* 2012, Schumaker *et al.* 2012, Tetlock 2007,

Vanstone & Finnie 2010, Yu *et al.* 2013].

2.2.2 "New News" and Twitter

There exists proof showing a dramatic evolution in the way news disseminates [Leinweber & Sisk 2011]. Around late 2007, there emerged a significant increase in news count, news depth (news stories per stock) and news breadth (coverage for lesser-known companies). The cause grew from enhancements that effectively provide journalists with electronic chunks of informative pre-news material. Thereby making the ability to write stories easier. The main result to come from this was a higher correlation in returns for event-driven trading strategies post 2007 [Leinweber & Sisk 2011]. The intensification of the dissemination process of news was called "New News".

"New News" is a concept that also encapsulates a technological innovation in how institutions handle releases. For instance, Dow Jones and Thomson Reuters produce machine readable news that directly integrates with trading software. Given this purview, Twitter is an interesting and highly-sought dimension. By using the Twitter API, investors are granted instant access to electronic messages that carry up to 140 characters of unstructured text. Twitter has developed into an instrumental source material for traders. It is for this reason that Twitter has systematized the use of a special character "$" to reference a stock. One instance would suggest carrying "$AAPL" upon referencing Apple Inc. Investors and industry are increasingly investigating Twitter-based strategies because tweets come under the heading of both "New News" and "Pre-News".

Pre-news consists of informative material that originates from a direct source. An example of this would be an SEC filing located on its site. Later, traditional media

would use information within that document to tell a coherent story. It should be noted that markets begin to move prior to the release of traditional news. In contrast, Twitter gives its source (companies, investors, traders, and the general population) the ability to reach a broad audience immediately. Thus, it has two advantages. First, source material, if predicted, can lead to an earlier entry point into a directional trade. Second, unlike traditional media, Twitter is not confined to the number of news stories or journalists it can have. Without such limitations, Twitter produces information on a larger breadth of companies. A breadth of coverage is important because studies have found a greater price impact from news on stocks with a small total dollar market value (Small-Cap) as compared to larger corporations [Leinweber & Sisk 2011].

The inability to differentiate news from no-news is the difficulty of using Twitter. By definition, traditional media presents newsworthy material. Contrastingly, Twitter has no guidelines for its correspondents. Up to this point, the solution in literature has been to examine collective sentiment from tweets [Bollen *et al.* 2011, Makrehchi *et al.* 2013]. One group studied how collective mood captured in public tweets (February 2008 - December 2008) impacts the Dow Jones Industrial Average (DJIA) [Bollen *et al.* 2011]. The Google-Profile of Mood States (GPOMS) software [Bollen *et al.* 2011] captures and separates mood types into the following: calm, alert, sure, vital, kind, happy. By running GPOMS, six time-series are generated to represent the mood states. The time-series dynamically capture daily mood changes. By using the "calm" time-series along with historical DJIA prices, researchers achieve an accuracy of 86.7% in predicting the DJIA's following day price direction [Bollen *et al.* 2011]. Results such as this do show the

potential for considering Twitter as a source for investment decisions. Although, it should be noted that while the DJIA is an important indicator for economists, it is not a tradeable asset. An obvious need is to go beyond general population sentiment into more company-centric prediction tasks.

Leveraging the Twitter standardized "$" is a common way to filter tweets about a particular company. Using this method, one can investigate how stock prices respond to the emotional valence (positive/negative sentiment) of tweets mentioning that stock [Sul *et al.* 2014]. Recently, by collecting and analyzing tweets mentioning companies from the S&P 500, results show that stock returns revealingly relate to their tweets' emotional valence [Sul *et al.* 2014]. Furthermore, tweets from users with a greater number of followers had an impact on same-day returns while users with fewer followers had an effect on 10-day returns [Sul *et al.* 2014].

Another thought is to try the reciprocal approach; that is label tweets according to significant market events [Makrehchi *et al.* 2013]. For example, label all tweets as "Positive" if they precede a 3% price increase. Similarly, label "Negative" tweets as those with wide negative swings. By building a corpus of labeled tweets, supervised machine learning techniques can be applied. Doing this, one learns which words in tweets can be evidence of forthcoming market events. The approach mentioned here was able to outperform the metric of precision by 8% when compared to a baseline model ("mood" words sentiment classifier) [Makrehchi *et al.* 2013]. Furthermore, a trading strategy from this model was able to beat the S&P 500 index by 20% over the four months examined [Makrehchi *et al.* 2013]. It should be noted that the 20% does not account for any

transactional costs.

2.3 Prediction Schemes

All prediction models discussed in this work originate from machine learning. The most common type of learning is supervised learning. The learning process ingests pre-labeled examples for training a prediction system. The designer teaches the learning algorithm by sending details about examples. Details, under this context, are called features. Examples refer to samples. Along with features, an algorithm is shown the output of each sample. During training, using only features, the algorithm predicts an output for samples. It is the designers job to maximize the quality of predictions. For this reason, the prediction is evaluated using an error function. Depending on the task, the output can be either a real-valued number or a class label. Regression schemes predict real-valued outputs while classification systems predict class scores. One optimizes a regressor or classifier by minimizing its prediction errors. As seen, general components for supervised learning are the following: feature set, samples, and labels.

2.4 Natural Language Processing

2.4.1 Bag-of-Words

Regardless of the news medium, a machine requires the ability to process unstructured text. Since it is easier to define rules and logic in a structured format, it is common to turn text to a structured schema. It is helpful to leverage preprocessing routines from natural language processing prior to prediction tasks. A review paper in text mining for market prediction highlights that nearly all studies use the Bag-of-Words model

Table 2.1: Bag of Words Features

amid	berkshir	buffett	elimin	exxon	firm	mobil	oilpric	plung	sell	share	stake	warren
1	1	1	1	1	0	0	1	1	0	0	1	1
0	0	1	0	2	1	1	0	0	1	1	0	1

[Nassirtoussi *et al.* 2014]. The model captures word frequencies in a document-term matrix scheme.

Several preprocessing methods are available in converting text to a structured schema. The basics are to tokenize and normalize text. Tokenizing separates words by empty spaces or regular expressions. Normalizing is made up of removing special characters, transforming words to lower case, stemming and/or lemmatization (reduce words to a simpler more common form), and removing stop words.

Bag-of-Words uses preprocessed common words for its feature set and a documents' word frequencies as sample features. The following two sentences are considered: "*Warren Buffetts Berkshire eliminates Exxon stake amid plunge in #oilprices.*", and "*Warren Buffetts firm sells Exxon Mobil shares #exxon*". The sentences are preprocessed (tokenized and normalized) as shown in Figure 2.2. For the two samples, Table 2.1 provides the feature set and document-term matrix.

Once in this schema, any linear or non-linear classifier may be used for prediction. The frequent choice for Bag-of-Words are linear classifiers, namely, Support Vector Machines (SVM) and Naive Bayes. Since the task considered is document classification (more specifically, sentiment classification), one must generate a vector of positive/negative labels for each sample.

Event Study research efforts differ in steps taken during preprocessing. Furthermore,

Example: Preprocessing pipeline for Bag of Words Model - Using NLTK software

```
>>> import nltk

>>> string1 = "Warren Buffetts Berkshire eliminates Exxon stake amid plunge in #oilprices."
... string2 = "Warren Buffetts firm sells Exxon Mobil shares #exxon"
```

Tokenize using NLTK tokenizer

```
>>> tokens1 = nltk.word_tokenize(string1)
... tokens2 = nltk.word_tokenize(string2)
... print ', '.join(tokens1)
... print ', '.join(tokens2)

Warren, Buffetts, Berkshire, eliminates, Exxon, stake, amid, plunge, in, #, oilprices, .
Warren, Buffetts, firm, sells, Exxon, Mobil, shares, #, exxon
```

Make text lower case and remove punctuation

```
>>> lowertokens1 = [word.lower() for word in tokens1 if word.isalpha()]
... lowertokens2 = [word.lower() for word in tokens2 if word.isalpha()]
... print ', '.join(lowertokens1)
... print ', '.join(lowertokens2)

warren, buffetts, berkshire, eliminates, exxon, stake, amid, plunge, in, oilprices
warren, buffetts, firm, sells, exxon, mobil, shares, exxon
```

Stem Words using NLTK Porter Stemmer

```
>>> stemmedLowerTokens1 = map(nltk.PorterStemmer().stem, lowertokens1)
... stemmedLowerTokens2 = map(nltk.PorterStemmer().stem, lowertokens2)
... print ', '.join(stemmedLowerTokens1)
... print ', '.join(stemmedLowerTokens2)

warren, buffett, berkshir, elimin, exxon, stake, amid, plung, in, oilpric
warren, buffett, firm, sell, exxon, mobil, share, exxon
```

Remove English Stop Words

```
>>> final1 = [word for word in stemmedLowerTokens1 if word not in nltk.corpus.stopwords.words('english')]
... final2 = [word for word in stemmedLowerTokens2 if word not in nltk.corpus.stopwords.words('english')]
... print ', '.join(final1)
... print ', '.join(final2)

warren, buffett, berkshir, elimin, exxon, stake, amid, plung, oilpric
warren, buffett, firm, sell, exxon, mobil, share, exxon
```

Create Feature Set for Bag of Words Model

```
>>> wordFeatures = list(set(final1) | set(final2))
... wordFeatures.sort()
... print ', '.join(wordFeatures)

amid, berkshir, buffett, elimin, exxon, firm, mobil, oilpric, plung, sell, share, stake, warren
```

Figure 2.2: Preprocessing Techniques in NLTK

there are many variants of SVMs and Naive Bayes. Thus, we look for answers in the field of sentiment classification. For sentiment in larger documents, SVMs outperform Naive Bayes whereas the reverse holds true for smaller documents [Wang & Manning 2012]. Furthermore, adding bigrams extends Bag-of-Words and enhances sentiment classification

performance [Wang & Manning 2012]. Using the example sentences from above, the phrase "sell share" would be a bigram (2-gram) feature. Finally, a combination of Naive Bayes and SVM called NBSVM achieves strong and robust performance across standard datasets (both large and small documents) [Wang & Manning 2012].

2.4.2 Distributed Word Representations

Simplicity and accuracy are the reasons why computational finance researchers select the Bag-of-Words model. It is, however, easy to see that words from the model are treated completely as symbolic. There is no notion of word relationships, ordering, or context. For example, in trading, considering the terms "buy" and "sell", the words "stocks", "shares", "options", "bonds" have a higher probability of being surrounding words as opposed to "laugh" or "cry". Word frequencies in Bag-of-Words will not find the relationships above. Newer trends in Textual Analysis, using distributed representations of words, may capture such similarities and associations.

The open-source implementation, word2vec, has gained acclaim due to its efficiency in learning quality representations of words in a continuous vector space [Mikolov *et al.* 2013a]. Word Embeddings map to a continuous n-dimensional vector. The ability to quantify relationships amongst lexicons is the benefit of storing words as vectors. The size of vector dimension needs scaling to handle an increasing dictionary of words [Mikolov *et al.* 2013a]. The word2vec tool can train over 1 billion words per day on a single machine and can be parallelized to scale up significantly [Mikolov *et al.* 2013a].

During training, word2vec uses the idea that words in similar context often appear together. The model fine-tunes word vector representations during learning. This is

achieved by looking at how well each word in a sentence can predict the likelihood of surrounding context specific words (skip-gram model)[Mikolov *et al.* 2013a]. In doing so, linguistic structure is found in word representations [Mikolov *et al.* 2013b]. Notably, word representations in a vector space can be grouped via similarity (cosine distance - 2.1). Provided is a vocabulary of 3 million words and phrases each represented by a 300-dimensional vector [Word2Vec 2016]. Presented here are examples of linguistic relationships using the above vocabulary (trained on Google News dataset - approximately 10 billion words).

Cosine Distance:

$$\cos(u, v) = \frac{u \cdot v}{\|u\|\|v\|} \tag{2.1}$$

Linguistic Relationship 1: Informal usage of vocabulary: Hydraulic Fracturing is a technique commonly used in the United States to extract oil. Media often refers to it as fracking. Table 2.2 represents the most similar word representations to "fracking" using cosine distance.

Linguistic Relationship 2: Synonyms and Antonyms: Syntactically synonyms and antonyms both fit well in similar sentences. Thus, their distributed word representations are often close. For example, "bullish" or "bearish" can both be used in "The market is (*insert word*) on this stock". The contextual similarity is shown in Table 2.3 using word representations closest to the adjective "bullish".

Table 2.2: Words Similar to "fracking"

word	cosine distance
hydraulic_fracturing	0.853974
hydrofracking	0.768026
Fracking	0.756967
hydro_fracking	0.725470
fracking_fluid	0.659324
hydraulic_fracking	0.647175
hydrofracturing	0.639748

Table 2.3: Words Similar to "bullish"

word	cosine distance
bearish	0.882593
bullishness	0.683763
overbought	0.673349
Bullish	0.657814
bearishness	0.654787
bullish_bias	0.642219
oversold	0.635528

Table 2.4: Word Analogies using Word Embeddings

china	*is to*	xi_jinping	*as*	zimbawe	*is to*	**robert_mugabe**
tehran	*is to*	iran	*as*	riyadh	*is to*	**saudi_arabia**
tesla_motors	*is to*	elon_musk	*as*	chrysler	*is to*	**sergio_marchionne**

Linguistic Relationship 3: Word analogies: Examining vector transformations on trained word embeddings leads to interesting properties. One can compute vector offsets (addition and subtraction) and then apply cosine distance on the resulting vector to find most similar words. The standard example shows how vector offsets capture gender regularities in linguistics. For word embeddings, the analogy "king is to man as queen is to ?" can be interpreted as $vector(?) \approx vector(king) - vector(man) + vector(queen)$. The closest vector to the result is the vector(woman). Word Embeddings capture relational similarities by preserving multiple aspects of similarity (e.g., gender and royalty) [Levy *et al.* 2014]. Thus, vector offsets aim to change some attributional similarities while keeping others the same [Levy *et al.* 2014]. Furthermore, multiplicative transformations on vectors improve performance on certain relational similarity tasks [Levy *et al.* 2014]. Table 2.4 contains three examples revealing the following relationships: (country, leader), (capital city, country), and (company, CEO). The resulting words in bold are calculated using the method from the gender example above.

In looking at the current state of financial Event Studies, no language model has used word embeddings. It is worthwhile to investigate this further as recent trends in sentiment classification move towards language models that use the full textual representation.

2.5 Time-Series Analysis

It is fair to use categorical values instead of continuous values to simplify classification tasks. For instance, instead of stock price, Up/Down can be used as labels to determine direction. Similarly, Buy/Sell/Neutral can be used as outputs for trading decisions. As it stands, most Event Study projects choose input features that are derived solely from Text. Thus, stock quotes alone determine categorical values.

Only a few research efforts have considered the influence of News and historical prices simultaneously. One compares accuracies for predicting stock direction on the following models: *Price*, *News*, and *Price and News* combined [Zhai *et al.* 2007]. *Price and News* combined achieve an accuracy of 70.1% while *Price* and *News* separately achieve 58.8% and 64.7%, respectively [Zhai *et al.* 2007]. Accuracy was simulated on BHP (stock symbol for an Australian company named Billiton Limited) using a set of articles from the Australian Business Review. Input features used technical indicators extracted from the closing price of BHP [Zhai *et al.* 2007].

An alternative to extracting price features, is to use previous day prices directly with textual features. In separate experiments, three days of historical prices were used. The three prices were used as additional features to text. Thus, increasing the accuracy of predicting next day price direction [Bollen *et al.* 2011].

To move beyond classification, discrete value predictions of a stock 20-minutes after a news release were tried [Schumaker & Chen 2009]. Unsurprisingly, to predict future prices, a baseline quote is required as one of the features [Schumaker & Chen 2009]. Less obvious, however, is that while the baseline price feature has a large weight, textual

features are undoubtedly needed to increase performance [Schumaker & Chen 2009].

In all cases seen, prices have been used only as additional features in a machine learning task. Furthermore, much of the history of prices have been ignored. These simplifications most probably occur due to the difficulty of properly integrating time-series components with feature based textual components. Nonetheless, there is enough evidence to highlight the need to investigate further techniques that incorporate a mixture of text and time-series.

2.5.1 Stock Returns

We look to common assumptions from Quantitative Finance for a more sophisticated analysis of financial time-series. The first and foremost undertaking is to choose an input representation for a stock. Therefore, let P_t be the price of a stock at time t. Since we are mostly concerned with the change in prices, we look to stock returns.

Simple Return: R_t [Tsay 2005]:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} = \frac{P_t}{P_{t-1}} - 1 \tag{2.2}$$

A statistical distribution aids in performing analysis (e.g., mean and variance) or modeling (e.g., forecasting). Thinking of stock prices as random variables, one first tries to rationalize that prices follow a normal distribution. Under this assumption, a stock price can be modeled using just the mean and variance. Since normal distributions

can take both positive and negative values, an inadequate assumption is made that prices can go below zero. Furthermore, the multiperiod simple return (2.3) is computed as the product of simple returns. Multiplying two or more normal distributions does not guarantee a normal distribution. Therefore, a multiperiod simple return would not follow a normal distribution and would no longer be able to be modeled by mean and variance. Given the above reasoning, a different assumption for prices is needed.

Multiperiod Simple Return: $1 + R_t[k]$ [Tsay 2005]:

$$1 + R_t[k] = \frac{P_t}{P_{t-k}} = \frac{P_t}{P_{t-1}} \times \frac{P_t}{P_{t-2}} \times \cdots \times \frac{P_t}{P_{t-k}} \tag{2.3}$$

Multiperiod Simple Returns do not account for compounding interest. Therefore, considering the use of compounded returns, instead of compounding at discrete intervals (e.g., monthly or annually), one can use continuously compounded returns (log returns 2.4). Log returns have nice properties that are mentioned below.

Continuously Compounded Return (Log Returns): r_t [Tsay 2005]:

$$r_t = \ln(1 + R_t) = \ln \frac{P_t}{P_{t-1}} = p_t - p_{t-1}, p_t = \ln(P_t) \tag{2.4}$$

In comparison to earlier, assuming stock prices have a lognormal distribution, stock prices can no longer be negative since lognormal distributions are bounded below by zero. Also, log returns infer a normal distribution. This is because if X has a lognormal distribution then $\ln(X)$ is normally distributed. A further advantage of using log returns is seen when computing multiperiod log returns (2.5). The resulting addition of two normal random variables is a normal random variable. Thus, multiperiod log returns follow a normal distribution. Finally, mean and variance can now be used to model simple returns, log returns, and multiperiod log returns.

Multiperiod Log Returns: $r_t[k]$ [Tsay 2005]:

$$r_t[k] = \ln(1 + R_t[k]) = r_t + r_{t-1} + \cdots + r_{t-k+1} \tag{2.5}$$

For prediction tasks, modeling prices directly is not always necessary. Stock quotes have simply been used as features. However, in time-series classification it may be better to use both current and past prices entirely. To do so, a full representation such as price shifts or log returns, can be used. Given an input representation for training, generalizing to unseen data is important. Generalizing becomes more difficult in finance because stocks differ greatly in the price at which they trade. To show this, Figure 2.3 compares share prices and log returns for two oil conglomerates (Chevron - CVX and British Petroleum - BP). In viewing stock prices ($110 versus $37), it is difficult to see any similarities

between the two companies. However, when referring to log returns, it is clear visually that the two oil companies are correlated. Thus, normalization is a valuable property gained in using log returns.

2.5.2 Similarity Measures

There are peculiarities in using temporal models. One example is the observation of the problem of clustering or classifying time-series. A common, but time-consuming, method is to extract features from time-series and use a feature-based model. Relevant features, however, differ from task to task. Therefore, we would like to examine classification and clustering using just the input representation. To perform such a task, we need to be able to identify a similarity between two or more time-series. If a similarity can be found then, clusters can be formed, and patterns can be classified. While many similarity measures have been proposed, we look at Euclidean Distance and Dynamic Time Warping. Euclidean Distance is chosen since it is the most used and simplest to interpret. Dynamic Time Warping is selected due to its state of the art achievements in time-series classification [Giusti & Batista 2013].

Euclidean Distance (ED)

Let X and Y be two time-series of equal length n.

Euclidean Distance:

$$ED(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.6}$$

where: x_i is the i^{th} value in X

y_i is the i^{th} value in Y

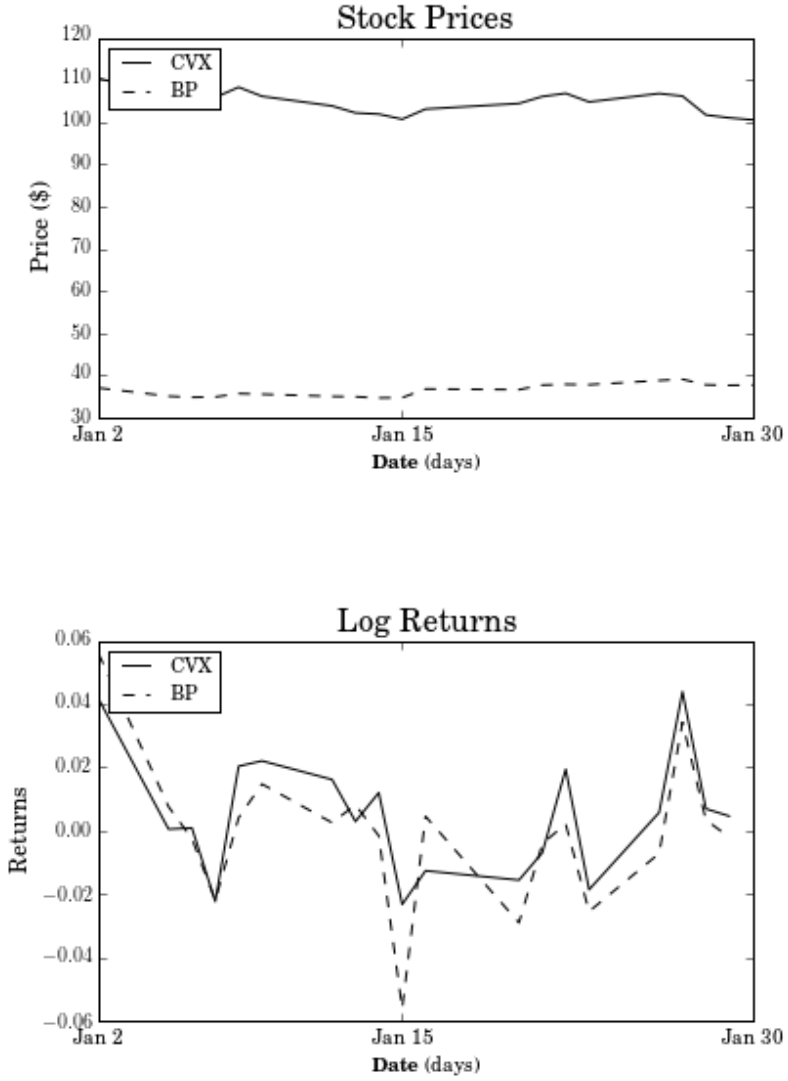


Figure 2.3: Prices versus Log Returns

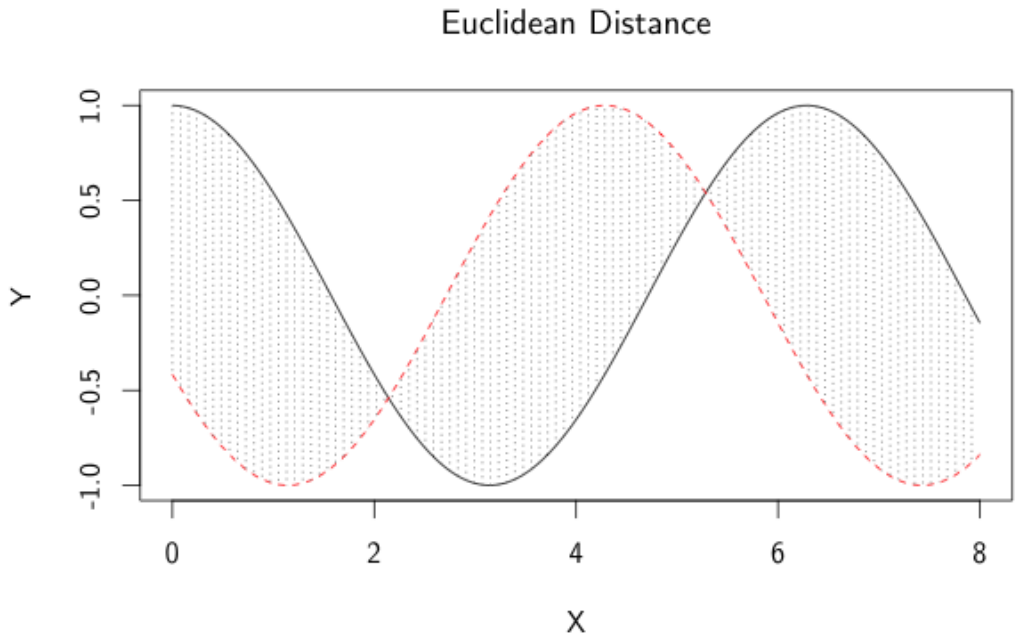


Figure 2.4: Euclidean Distance

Figure 2.4 illustrates the ED between two time-series generated from the cosine function. It is simply the sum of the point to point distances along the x-axis. Because of the point to point constraint, it is impossible to calculate the ED of two series of varying length. Furthermore, for the example, it is worth noting that the two are equivalent with a phase shift. Since the ED calculations are fixed in time, they are unable to capture the equivalence. Dynamic Time Warping, on the other hand, is a similarity measure that can compress and stretch time to match points in space.

Dynamic Time Warping

Dynamic Time Warping (DTW) uses a dynamic programming approach to finding a composition of two time-series that minimizes the distance between them [Berndt & Clifford 1994].

Let X and Y be two series of lengths m and n respectively.

Let (i, j) be coordinates captured in an n by m grid.

Let $W = w_1, w_2, \cdots, w_k$ represent a path of points along the n by m grid.

Dynamic Time Warping [Berndt & Clifford 1994]:

$$\begin{aligned} DTW(X,Y) &= \min_w \Big[\sum_{k=1}^{P} \delta(w_k) \Big] \\ \text{where: } w_k &\text{ is point } (i,j)_k \text{ from Path P} \\ \delta(w_k) &= (x_i - y_j)^2 \\ x_i &\text{ is the } i^{th} \text{ value in } X \\ y_j &\text{ is the } j^{th} \text{ value in } Y \end{aligned} \tag{2.7}$$

A comparison of 48 similarity measures across 42 time-series data sets, found that DTW significantly outperformed all other measures [Giusti & Batista 2013]. Still, DTW is yet to be used in finance. The reason is that the complexity of DTW is $O(m^2)$. Unfortunately, in trading, speed is of utmost importance. In 2012, a team won the best paper award (SIGKDD) for significantly increasing the rate of DTW calculations

[Rakthanmanon *et al.* 2012]. In using one billion data points, time comparisons were computed between the UCR-DTW and a previous state of the art DTW implementation. UCR-DTW took 1.83 minutes to finish while the previous model took 38.14 minutes [Rakthanmanon *et al.* 2012]. Moreover, for the same data points, UCR-DTW beat a state of the art ED model (2.40 minutes). Thus, increasing the speed as needed in finance. The authors offer a link to the source code for the UCR-Suite [UCR-Suite 2016].

2.5.3 Classification

In terms of classifiers, time-series representations do not have a plethora of options. K-Nearest Neighbors (K-NN) is commonly favored. It is the norm because a similarity measure is the only requirement. The algorithm is simple. To classify an input series, first a similarity measure is used to compare against all representative series in the training set. Then, it is classified using the most common label from the K most similar series.

An alternative method is to use regression schemes to convert real-valued numbers into a class label. A conversion grants access to many regression algorithms for time-series predictions. As an example, we consider a regression model that predicts stock price movements in cents. All we need for conversion to classification is a buy criterion. If the prediction is greater than x cents, then buy, contrarily do nothing. The downside of the approach is that the training optimizer will only be concerned with minimizing prediction error on real-valued outputs, not buying decisions. The upside is that it allows usage of conventional methods such as Linear Regression, Ridge Regression, Support Vector Regression, and Neural Networks. The strategy of leveraging regression systems for a classification task is considered in this work.

Chapter 3
Research Methodology

3.1 Research Hypotheses

This work is based on three related research hypotheses.

3.1.1 Identifying Events on Twitter

Hypothesis 1: *In order to move beyond general group sentiment on Twitter, one needs to determine how to use tweets similar to news headlines. Thus, for tweets, keyword-based similarity clusters will generate Twitter events that resemble headlines.*

As mentioned, filtering irrelevant tweets from newsworthy tweets is a difficult task. This is the reason why most researchers have filtered noise by averaging sentiment through a large group of tweets. Aside from using the "$" symbol to signify stocks, individual news events are commonly ignored. Luckily, with the sheer number of users and tweets, related messages arriving in a small time interval can aid in forming events. To better understand the intuition, considering an oil spill in Mexico, it is likely that clusters of tweets will pour in for a short duration mentioning "*oil*", "*spill*", and "*mexico*". A procedure for event identification is presented later.

3.1.2 New-News to fix Pre-News & Lagged-News Effect

Hypothesis 2: *Given Twitter events, it is possible to address Pre-News and Lagged-News concerns.*

One recurring result in prior research has been the stock prices moving prior to the release of news. In comparison, Twitter often discovers new stories faster because individuals post eye-witness accounts. Of course, one caveat is that statements are not verified. Still, as markets move on rumors and truths, it is possible that using Twitter will reduce the amount prices move before articles. Secondly, roughly all studies ignore how to deal with lagged-news. Observing again an oil spill in Mexico, at first, breaking the story may negatively affect stock prices. Many times there will be follow-up articles explaining how prices fell after news of an oil spill. These lagged articles will not have the same impact on markets as the original story. However, a learning algorithm will most likely be trained to classify it as a new negative event. Thus, an emphasis is placed in this work to both identify news and to reduce lagged-news.

3.1.3 Language and Time-Series Ensemble Models

Hypothesis 3: *When using modern methods on text and prices (in their original representation), the performance of a trading strategy – on Twitter generated events – that predicts stock price jumps can be profitable.*

For language models, many researchers have not gone beyond the Bag-of-Words model.

Top NLP tools from outside of the Event Study community will be discussed later. Furthermore, when using stock prices, Event Study models reduce the dimensionality of the input space by extracting a limited number of price features. Recent trends in machine learning aim to reduce the time spent on feature engineering. Representation learning and deep learning groups are leading this trend. We use prices in their full representation. Later sections highlight time-series models solely based on temporal data, and how to independently ensemble language and time-series models.

3.2 Approach

Both general and company-centric news influence traders. As such, we have designated two datasets; a set of general tweets, and one that explicitly mentions companies. Selecting a particular industry avoids the additional task of topic modeling. Thus, the energy sector is the subset of tweets for use when there is no specifying of individual companies. The choice is due to the sectors' mass appeal in media and trading. In reverse, when tweets single out companies using Twitter's "$" character, we filter stocks by using a static list of the S&P 500. The approach allows for an examination of differing styles of tweets.

Assigning the financial security associated with a tweet is straightforward when the symbol is in the text. However, a difficulty lies in connecting energy events to tradeable financial securities. A potential solution is to use an Exchange-Traded Fund (ETF). The design of an ETF is such that it holds positions in financial investments being a tracker for an index or a given sector. An ETF is a tradeable investment fund. There are many

ETF's that track the energy sector. By using an ETF, and stocks from the S&P 500, we now have access to run Event Studies on both datasets.

For tweets using the "$" sign, it is immediately clear that storing any tweet that mentions a company is all that is needed. However, it is unclear without domain knowledge how to collect energy-related tweets. Now we focus on how to solve the lack of domain expertise. Leveraging data analysis and Twitter allows the automation of gathering domain specific knowledge. As a consequence, the requirement of having energy sector experts is removed. Later sections describe the procedure for amassing domain expertise. Provided here is a list of steps required to study price influence of events in the energy sector. While the energy sector is specifically mentioned, the approach is general and can be used for any other sector.

High-Level Overview:

1. Mine Twitter API to identify relevant individuals as topical experts.

2. Examine language from tweets of topical experts to find focal keywords within energy domain.

3. Track and store all tweets mentioning company-centric stock symbols or mined energy keywords.

4. Provide a method to define "New Event" from a cluster of similar tweets.

5. Develop a procedure to calculate a numerical Event Matrix that aligns events with financial time-series.

6. Construct language models from tweets that best classify price spikes.

7. Construct a temporal model that classifies large price movements using share prices.

8. Measure the performance of language, temporal, and ensemble models on the validation and testing data.

9. Use classified outputs from prediction model as buy/sell trading signals.

10. Evaluate trading metrics (Returns, Sharpe Ratio, Profit Per Share).

3.3 Dataset Partitions

A significant concern for predictive financial models is overfitting; the lack of ability to generalize. We alleviate the suspicion above to some extent by holding out large sections of our dataset for validation and testing. Supervised learning algorithms learn by seeing examples from the training set, then predict on unseen data. The dataset partition consists of a training, validation, and testing set. Choosing two longer-term hold out periods is necessary to avoid bias from short-term market conditions. The validation set, while being unseen by the learning model, allows parameter tuning. Intraday trading ensures a considerable volume of data, permitting the ability to partition higher percentages. Unlike other applications, finance does not grant shuffling of data due to its sequential nature. Therefore, the sets are arranged in a preserved order of trading days. Also, partitioning using trading days avoids having data from the same intraday period between two sets. The dates used for partitions are described next.

Table 3.1: Statistics of Twitter Identified Energy Experts

Tweet Tracking Dates	03/15/2015 - 08/01/2015
Number of Tweets	98,038
Number of Unique Tweets	78,723
Number of Users	95

3.4 Data

There are two forms of collected data, namely, tweets and stock quotes. The choice of trading frequency determines the level of detail in constructing an Event Study. Tweets measure timestamps to the second. Therefore, we may choose a trading rate of either minutes or days. Since intraday trading provides a larger volume of data; we elect minutes. The Twitter API is used to retrieve tweets. MongoDB is then used to store tweets for analysis. This section provides a summary of curated data. The descriptions are separate for both datasets; general and company-specific.

3.4.1 Energy Sector Tweets

Twitter allows users to follow other users. Users on Twitter are considered popular if they have a large number of followers. A preliminary investigation collects all tweets from popular users who focus on the energy sector. The analysis of the set of energy users is conducted to generate domain knowledge. Table 3.1 summarizes the collection of tweets used to mine energy-related keywords. A later section details how to curate the collection.

After identifying keywords to track for tweets, we ensure that our financial data match. Fidelity's stock screener is utilized to capture the largest and most traded companies from

Table 3.2: Statistics of Financial Time-Series

Trading Frequency	Minute
Number of Energy Stocks	171
Market Capitalization	>= $1,000,000,000 (USD)
Volume (90 Day Average)	>= 250,000 shares traded

Table 3.3: Statistics of Energy Sector Tweets used for Language Models

Tweet Tracking Dates	11/15/2015 - 03/01/2016
Number of Tracked Keywords	45
Number of Tracked Symbols	171
List of Keywords	Provided in Table 4.4

the energy sector [Fidelity 2016]. The filters in use are Market Capitalization and the 90-day Average of Volume Traded. The chosen filters assure that selected stocks are liquid (easily traded). Illiquid stocks could have high transaction costs. Thus, high transaction costs would be hard to justify given the underlying assumptions used in trading strategies. The full list of energy stocks is included in Table A.1. Table 3.2 summarizes the financial data.

Language models for general events are trained and tested on tweets mentioning energy keywords. Table 3.3 summarizes the collection.

We evaluate the Event Study over a period of approximately four months. The timeframe does not include the additional four months used to find top energy-related keywords. Table 3.4 displays the time to be spent on training, validating, and testing all learning algorithms. Predictions from validation and testing are also of use for measuring the performance of trading metrics.

Table 3.4: Energy Sector Evaluation Period

Training Period	11/15/2015 - 12/31/2015
Validation Period	01/01/2016 - 01/31/2016
Testing Period	02/01/2016 - 03/01/2016

Table 3.5: Statistics of Company-Centric Tweets used for Language Models

Tweet Tracking Dates	12/21/2015 - 05/01/2016
Number of Tracked Symbols	396
List of Symbols	Provided in Table A.2

3.4.2 Company-Centric Tweets

The second collection only stores a tweet given the presence of the "$" character. For example, any text containing "$FB" (symbol for Facebook, Inc.) is immediately saved into the MongoDB database. Twitter limits the number of tracked terms to 400. As such, we generate a list 396 symbols to track by using a static snapshot of the S&P 500. Table 3.5 arranges statistics on company-centric tweets. Table 3.6 includes the training and evaluation periods for the dataset.

Table 3.6: Company-Centric Evaluation Period

Training Period	12/21/2015 - 02/28/2016
Validation Period	03/01/2015 - 03/31/2016
Testing Period	04/01/2016 - 05/01/2016

3.5 Event Study Horizon

The horizon is established to measure the effect Twitter events have on prices. The relevant definitions are the event-time, pre-event region, and post-event region. As suggested, intraday impacts are of interest. Because of this, the event-time can be any minute during market hours that produces a complete post-event region.

To compute the financial metrics, the entry, and exit of trades must be clear. A simple exit strategy of closing a position after a set amount of time works well with Event Studies. The approach is to examine exit strategies of 30, 60, 120, and 240-minutes. As such, given time of event T, the post-even regions will be $(T, T + 30]$, $(T, T + 60]$, $(T, T + 120]$, $(T, T + 240]$. These intraday regions help differentiate how long the influence remains on stock prices.

Time-series models use the pre-event region for predicting movements. A larger time-span grants more input data for the forecasts. However, it reduces the number of events possible due to missing data. Since the markets open at 9:30 am EST, if an input requires 60-minutes of financial quotes, then it will not allow a trade before 10:30 am EST. Accordingly, a 30-minute selection defines an area of $[T - 30, T)$.

Experiments show evaluation on holding periods of 30, 60, 120, and 240-minutes. Events compose of studies at one minute frequency intraday. Since language models do not require financial quotes from a pre-event region, the dataset is a superset of the time-series dataset.

3.6 Evaluation Metrics

The selected evaluation metrics are price spike accuracy (with a focus on precision/recall), and trading metrics. Directional accuracy, representing a binary classification with UP/DOWN as classes, is a consistent choice amongst researchers because it displays the algorithms' ability to predict price direction. It is, however, less important by industry standards since trading only on price direction can lead to financial losses. For example, if incurred transaction costs are higher than stock gains then the overall result is negative. Incurring high transaction costs is a common pitfall in designing trading strategies. The binary classification of UP/DOWN also does not follow trading logic; most of the time it is best to make no trade at all.

While similar to directional accuracy, the ability to predict price spikes can have considerable benefits. Let us define a price spike as stock movement greater than $|x|\%$ within time t. We can empirically find the value for x. It need not be symmetric, in such a case one can use two positive values x and y. The values must be large enough to offset transaction costs, but small enough to find enough events. To view this as a classification problem, we define the three classes of POSITIVE/NEUTRAL/NEGATIVE.

$$\text{CLASSES} = \begin{cases} \text{POSITIVE} & : +x\% \text{ movement in time } t \\ \text{NEUTRAL} & : [-y, x]\% \text{ movement in time } t \\ \text{NEGATIVE} & : -y\% \text{ movement in time } t \end{cases}$$

Given this setup, we can train a prediction classifier that works well with classical

buy/sell trading logic: purchase the stock when the classifier predicts POSITIVE; short the stock when the prediction is NEGATIVE; and do nothing for a NEUTRAL prediction. Lastly, the simplest exit strategy is to close the trading position after time t.

Precision/Recall [Davis & Goadrich 2006]:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.1}$$

where: TP: POSITIVE prediction is correct (True Positive)

FP: POSITIVE prediction is incorrect (False Positive)

FN: NEGATIVE prediction is incorrect (False Negative)

The aforementioned formula demonstrates how precision and recall are used in binary classification. For the price spike classification task we can consider a one versus all strategy. When measuring precision for the POSITIVE spike class, both NEUTRAL and NEGATIVE would be the incorrect category. Now, less formally, we describe precision and recall on the POSITIVE spike class. Precision answers the question "*When the classifier predicts a POSITIVE spike, how often is it right?*". Similarly, recall answers "*What fraction of all POSITIVE spikes does the classifier predict?*".

The emphasis on precision and recall stems from trading logic. To explain this, we look at cancer diagnosis. Normally, when diagnosing cancer, it is better to be safe than sorry. Thus, to not miss diagnosing cancer patients, it is better to trade off precision for recall. In contrast, trading does not have a huge penalty for missing out on buying opportunities. Profit and loss are only affected by precision. It is best to tune model parameters to maximize precision. The only consideration needed is to increase recall if there are not enough trading opportunities.

Formalizing the expected value of profits helps understand this reasoning. Here is a list of definitions and the Expected Value calculated for trading strategies with long only positions.

Expected Value for Trading Strategy:

Definitions:

P: Profit, where P = Price Bought - Price Sold

P_{Win}: Average \$ return on wins

P_{Loss}: Average \$ return on loss

$\Pr(Win)$: Probability of Win

$\Pr(Loss)$: Probability of Loss

Q: Quantity

TC: Transaction Costs

$$E(P) = (P_{Win} - TC) \cdot Q \cdot \Pr(Win) + (P_{Loss} - TC) \cdot Q \cdot \Pr(Loss)$$

$$E(P) = Q \cdot (P_{Win} \cdot \Pr(Win) + P_{Loss} \cdot \Pr(Loss) - TC \cdot (\Pr(Win) + \Pr(Loss))) \quad (3.2)$$

Note: $\Pr(Win) + \Pr(Loss) = 1$

$$E(P) = Q \cdot (P_{Win} \cdot \Pr(Win) + P_{Loss} \cdot \Pr(Loss) - TC)$$

When using classification predictions for trading:

Win = TP (True Positive)

$Loss$ = FP (False Positive)

$\Pr(Win)$ = Precision

$\Pr(Loss)$ = 1 - Precision

$$E(P) = Q \cdot (P_{TP} \cdot Precision + P_{FP} \cdot (1 - Precision) - TC)$$

$$E(P) = Q \cdot (Precision \cdot (P_{TP} - P_{FP}) + P_{FP} - TC)$$

Now that we have found a relationship to measure expected returns' we look at a concrete example. Given that TC is 0.1 units and $Precision$ is 0.5, what is the break even point for a strategy?

$$\begin{aligned} 0 &< Q \cdot (0.5 \cdot (P_{TP} - P_{FP}) + P_{FP} - 0.1) \\ 0.1 &< 0.5 \cdot P_{TP} + 0.5 \cdot P_{FP} \\ \frac{0.1}{0.5} &< P_{TP} + P_{FP} \\ 0.2 &< P_{TP} + P_{FP} \end{aligned}$$

$\therefore$ The strategy is profitable as long as the combined average returns are greater than 20 units. Optimizable parameters are precision and average returns.

The final evaluation criteria is achieved by computing trading metrics. The following industry standards are used as metrics: Returns, Sharpe Ratio, and Profits Per Share.

Sharpe Ratio [Sharpe 1994]:

$$\begin{aligned} \text{Sharpe Ratio} &= \frac{r_p - R_f}{\sigma_p} \\ \text{where: } r_p &\text{ is return of the portfolio} \\ R_f &\text{ is return of risk free rate} \\ \sigma_p &\text{ is standard deviation of portfolio} \end{aligned} \tag{3.3}$$

Sharpe Ratio is a far more telling metric than returns because it measures returns with respect to risk. As such, a strategy that produces 8% returns with very low volatility is much better than one with high fluctuations. The goal is to attain positive Sharpe Ratios. In general, a strategy with a Sharpe Ratio above three will be profitable daily [Chan 2008]. Due to the low-risk levels, strategies with lower average returns but larger Sharpe Ratios can be leveraged to maximize profit. An important task is to make certain that all values of risk and returns are annualized. With current economic conditions, the risk-free rate can be ignored. As of July 4th, 2016, the annual Treasury yield of a One-Year Government bond is 0.44%. This value will insignificantly affect Sharpe Ratios once trading returns are annualized.

Commission costs are often calculated on a per share basis. For example, a brokerage can charge a fee of 0.0075 cents per share on a trade. For this reason, we highlight the average profit per share in cents for our prediction algorithm. Doing so provides portfolio managers with a better view of how to associate their fee structure. Since there are different fee structures, we do not include transaction costs in our study.

Furthermore, we hope that the emphasis on precision and recall can aid in optimizing prediction algorithms for finance. Lastly, similar to some studies, providing trading metrics help bridge the gap between academia and industry.

3.7 Models

Models predict using input from two representations; text and financial time-series. Thus, all systems are either language models or time-series models.

For time-series, the task is not to compute long-term forecasts on a single stock. Thus, we do not consider traditional time-series forecasting designs such as autoregressive moving averages or stochastic models. Instead, the task compares machine learning techniques to mine data for patterns in events. Traditional models are applied: Linear Regression, Kernel Ridge Regression, Support Vector Regression, and K-Nearest Neighbors (with DTW similarity measure). Furthermore, with the recent trends of Deep Learning, we create systems using Recurrent Neural Network architectures. Further detail is provided in later sections.

Language models try to classify stock movement given the text within tweets. A later section explains the choices of models. The models for examination are NBSVM, RNNLM, and the Paragraph Model [Mesnil *et al.* 2014].

3.8 Model Ensemble

In attempts to attain greater results, combinations of models are studied. A combination of machine learning models is called an ensemble method. Our individual models belong to either a regression or classification category. Each time-series model except the K-Nearest Neighbors is a regression model. Every textual scheme predicts classes.

First, we consider joining regression models. Regression predicts real-valued outputs. Thus, a natural solution applies weighted averages between models. The approach scales with increasing predictors. Weights are computed with the goal to raise Annualized Sharpe Ratios. The validation set is used to approximate the ideal weights. The reason for approximation is to reduce the computational expense.

Table 3.7: Example: Ensemble Weights for Regressors

Model	M1	M2	M3
Sharpe Ratio (Validation)	1	2	1
Inferred Ensemble Weights	0.25	0.5	0.25

Models that individually perform better are given a higher weight in an ensemble prediction. Therefore, a weighted arithmetic mean of Sharpe Ratios determines the influence of individuals in a combined model directly. Table 3.7 provides an example of how three models' Sharpe Ratios contribute to their ensemble weights.

As previously mentioned, the metrics of interest are accuracy, precision, recall and financial statistics. These must all be calculated on trading decisions, not on regression outputs. Thus, even though regression models output a numerical value, they are then converted to a labeled class (BUY/NEUTRAL/SHORT). Classification models already predict those categories.

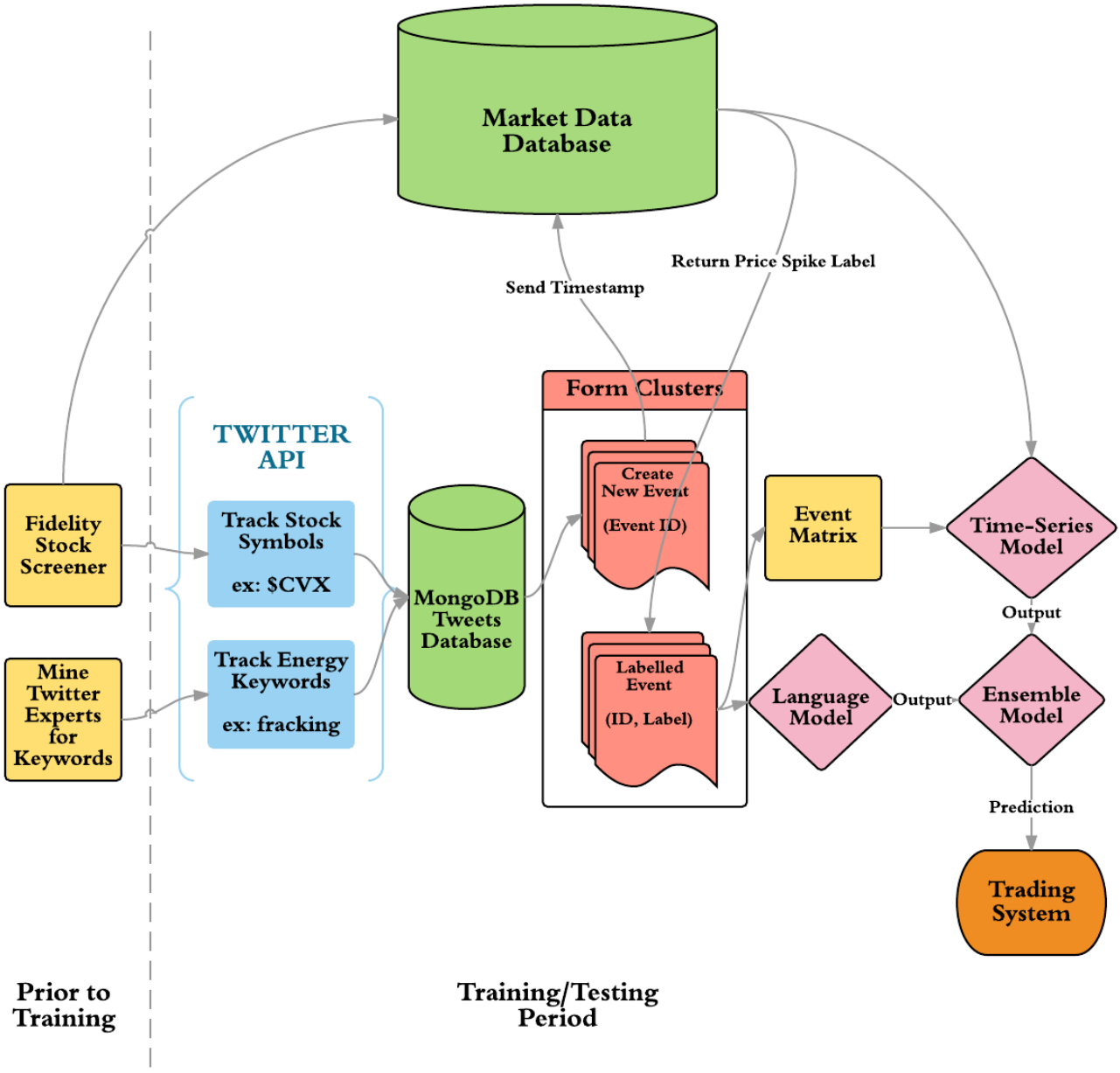
Since classification tasks predict labels, it is simplest to take the most common predicted class when building an ensemble. This is the strategy used on language models. If multiple learning algorithms conflict in a tie, then it is safest to select NEUTRAL.

3.9 Overview Diagram

Figure 3.1 presents the system design. The overview flows through the full procedure, starting from getting the list of stocks from Fidelity to the evaluation of trading decisions. A later section describes with detail all components from the system overview.

The system overview highlights the following components: Data Retrieval, Data Storage, Cluster Formation, Model Building, and Trading Metrics. Prior to training,

the data needed for retrieval are deduced. The two components of data are financial time-series and tweets. Tweets are retrieved using the Twitter API while financial data are gathered using QuantQuote [QuantQuote 2016]. The retrieved data are stored in MongoDB databases [MongoDB 2016]. Once all data are stored, clusters are formed as described. Clustered events and Market data are converted to a format suitable for classification tasks. The outputs from learning algorithms are ensembled and used as trading decisions. Figure 3.1 visualizes the procedure for constructing the proposed Event Study.


Market Data Database
Send Timestamp
Return Price Spike Label
Form Clusters
TWITTER API
Fidelity Stock Screener
Track Stock Symbols
ex: $CVX
Mine Twitter Experts for Keywords
Track Energy Keywords
ex: fracking
MongoDB Tweets Database
Create New Event
(Event ID)
Labelled Event
(ID, Label)
Event Matrix
Time-Series Model
Output
Language Model
Output
Ensemble Model
Prediction
Trading System
Prior to Training
Training/Testing Period

Chapter 4
Experimental Design

4.1 Mining Twitter Experts

Lacking domain expertise can be a hindrance for machine learning researchers. Also, NLP-related projects frequently require customized domain specific dictionaries. For Event Studies, both of the obstacles above can be lessened by mining knowledge using data analysis. The foundation for gaining expertise in the energy sector is described in this section.

Using the Energy Sector, the following exercise demonstrates how to mine for expertise on Twitter. With a Twitter account @OilTracker1 created, we select 52 users that seem to be chronicling the Energy Sector. Then using two Python libraries (Pandas and the tweepy API), we run an analysis on all 52 users. Twitter gives information that can be used to determine the popularity of a user. Table 4.1 shows popularity statistics for the top five hand-picked users.

Table 4.1: Statistics of Top 5 Users Ranked by # of Followers

Username	# Followers	# Friends	# Statuses	# Favorites
OGJOnline	93948	1101	10486	429
PlattsOil	57661	637	47715	141
IEA	56995	5213	9372	438
WorldOil	51403	481	4054	15
Rigzone	43106	475	6874	13

It should be noted that up until now we have only hand-selected users to follow. Thus, listed here is a procedure that uses tweepy to find the most influential Energy related Twitter accounts. The procedure uses the reasoning that networks for users in a sector will be inter-connected.

1. For the 52 users, collect the user ID for each Friend that they have on Twitter.
2. Construct a set called New Friends; defined as the union of all Friends mined in Step 1.
3. Calculate frequency counts to determine how many of the original 52 users follow the New Friends.
4. Sort New Friends by the largest number of common followers from Step 3 - where the maximum number possible is 52.
5. Run summary statistics on shared followers to determine a threshold for adding New Friend.
6. Optional: Retrieve profile description to confirm the type of user.
7. Add users above the selected threshold.

Table 4.2 shows results of Step 5. Given the numbers, here is the procedure used to update the experts list.

- A threshold of 20 is used to add a total of 67 new friends.

Table 4.2: Common Follower Statistics of Mined Twitter Accounts

Total Number of Users	35638
Number of Users with only one Common Follower	28455
Number of Users with at least 5 Common Followers	1431
Number of Users with at least 10 Common Followers	410
Number of Users with at least 15 Common Followers	169
Number of Users with at least 20 Common Followers	67
Number of Users with at least 25 Common Followers	32
Number of Users with at least 30 Common Followers	10

- To remove hand-picking unpopular friends, we remove 24 users from the original 52 that do not meet the threshold.

- Thus, @OilTracker1 follows 95 energy related Twitter Accounts.

Three profile descriptions of mined accounts are provided.

1. "*@WSJ senior energy reporter. Author of The Boom: How Fracking Ignited the American Energy Revolution and Changed the World.*"

2. "*Raising awareness about the economic and environmental benefits of clean, abundant and affordable #natgas.*"

3. "*The latest energy news from the Financial Times. Our customer service team is @FTcare.*".

The descriptions of the above profiles as well as the remaining 64 lead to believe that the procedure was successful in finding experts.

4.2 Knowledge Based Keywords

The Twitter API limits the public to a random 1% of all public tweets per day. While this is a large sample of approximately 5 million tweets per day, it would not suffice for Event Analysis. However, Twitter does provide the ability to track specific keywords. For this reason, an emphasis is placed on preplanning requirements for data collection.

To find keywords related to the Energy sector, we analyzed word frequency counts on experts' tweets. A total of 98,038 tweets were retrieved from March 1st, 2015 to July 31st, 2015. One feature in Twitter is the ability to retweet a different author's message. It is important to remove retweets as they refer to unoriginal content. Furthermore, duplicate messages add bias on word counts. A total of 78,723 original documents remained after removing retweets.

Words are separated using a highly developed Part-of-Speech (POS) tagger [Owoputi *et al.* 2013]. The chosen POS tagger was trained originally to handle Twitter's intricacies. Grammatical properties help determine classes of words that are more pertinent to the Energy sector. Table 4.3 shows the 10 highest occurring words for common nouns, proper nouns, verbs, hashtags, and adjectives. It becomes obvious that using verbs and adjectives has no place due to their general use.

Frequency counts are updated by combining seemingly relevant common nouns, proper nouns and hashtags (without the # symbol). Unfortunately, it is not reasonable to use the top occurring words as is. This is due to the generality of usage of certain words. For example, "u.s." and "china" can be used in many contexts. Therefore, the top occurring words were sorted and selected with the criteria of staying on-topic. Table 4.4 displays

Table 4.3: Top 10 Word Frequencies for each Part-of-Speech Tag

Rank	Common Noun	Proper Noun	Verb	Hashtag	Adjective
1	oil	us	is	#energy	new
2	gas	u.s.	are	#natgas	more
3	energy	china	be	#oil	natural
4	reports	platts	will	#aapg	crude
5	prices	lng	has	#news	global
6	production	mdn	says	#gas	first
7	industry	iran	have	#oilandgas	big
8	deal	fuelfix	can	#lng	top
9	power	north	could	#capitolcrude	next
10	market	uk	was	#fracking	latest

Table 4.4: Mined Keywords for the Energy Sector

aapg	barrel	capitolcrude	carbon	chevron
climate	coal	crude	crudeoil	drilling
earthquake	electricity	emissions	energy	epa
ethanol	exploration	exports	fracking	fuel
fuelfix	gas	gasoline	iea	keystonexl
lng	marcellus	mdn	natgas	offshore
oil	oilandgas	onshore	opec	petroleum
pipeline	platts	power	production	renewables
rig	shale	shell	solar	wind

45 selected words to track going forward.

The expertise shown by the list of selected keywords seem promising. To elaborate, we examine a few of the abbreviated words. LNG is a short form for "Liquefied natural gas". AAPG stands for "American Association of Petroleum Geologists". Fuelfix is a Houston-based company reporting on Energy news. MDN is short for "Marcellus Drilling News". Furthermore, the frequency counts of "oil", "energy", and "gas" are 12,570, 7,841, and 7,463, respectively. It is good news that a wide net of tweets are found by only a few words. With the list of keywords, we no longer are restricted to only following the

experts. Any post on Twitter mentioning the words listed in Table 4.4 will be retrieved.

4.3 Event Clusters

There are two types of tweets being tracked. The first set contains tweets that use "$" to refer to companies while the second set follows energy keywords. One tweet may not satisfy the requirements of being noteworthy. However, a burst of tweets on the same topic would assert likelihood. The following algorithm is put forward to cluster events.

1. For incoming tweets, tag text using POS tagger.

2. Compare tagged common nouns, urls, proper nouns and hashtags with prior events. If no such event has occurred in the past T minutes, create Event ID and map POS tags to a new event. If POS tags match with a past events' tags, then append tweet to closest old event. Do not use energy keywords in matching criteria.

3. Close an event if it does not have a new tweet appended to it within T minutes.

4. Save an event as noteworthy if the required number of tweets cluster within T minutes.

5. Once an event is created, save it for reference in merging future indistinguishable events.

6. Mark an event as lagged if it shares high similarity with an older saved event.

7. Convert clustered event time to a tradeable market time.

Table 4.5: Parameter Set for Clustering Events

Cluster Size for Energy Events	5
Cluster Size for Company-Centric Events	3
Time Limit for Clusters	60 minutes
Time Limit for Lagged Events	4 hours
Matching Criteria	2 words
Merging Criteria	3 words

8. For each clustered event, send timestamp to Market Data database to connect financial data. If a company is mentioned using the "$" then use the specific stock price. If the tweet is general, then use an energy ETF for price reference.

9. Use Step 8 to label price spikes as POSITIVE/NEUTRAL/NEGATIVE.

The suggested algorithm for clustering events has parameters. During the training period, the parameters may be empirically adjusted to improve performance. However, the event clustering algorithm is the most computationally expensive algorithm in this work. An extensive amount of parameter tuning could further improve results. The clustering method needs to identify events early enough to counter the pre-news effect. Also, merging handles lagged news by determining whether late tweets have an additional impact. Thus, the current parameter set is an initial solution for event clustering. Table 4.5 specifies crucial paramaters.

General tweets arrive at greater frequency than company-centric tweets. As such, for company-centric and energy tweets, the options for classifying a cluster are three and five tweets, respectively.

Further consideration is given to the handling of general tweets. Energy stocks will

be affected differently by common events. Thus, during training, we consider the price correlation of selected ETFs to energy events.

Once the cluster reaches the accepted size, the timestamp of the last tweet is used. We convert this timestamp to an acceptable market time for trading. Since Twitter timestamps appear at a second frequency, we round up to the nearest minute. It is important not to round down, as it would have a look ahead bias of seconds.

Rounding up to the next minute does not solve the scenario where exchange hours are not open. To deal with this, any event that comes during outside hours takes the next available market time. This strategy puts to use all newsworthy events leading up to the market opening at 9:30 am EST.

The lagged-news effect is dealt with by storing any noteworthy event for 4 hours; purging any incoming event with a strong resemblance.

A primary benefit of using the enumerated method is the ability to label tweets automatically. It is essential for supervised learning.

4.4 Model Type: Language Models

The purpose of the language model is to predict a price spike given a tweet. Since features are textual, there are no issues with missing values. Models use pre-processing techniques to normalize text. Two approaches are attempted for model selection. First, staying consistent with past studies, a Bag-of-Words style approach where words are symbolic, is used for features. Second, more recent models are tested using words converted to numerical vectors.

Table 4.6: One-vs-All approach for Language Model Predictions

Model	Sample 0	Sample 1	Sample 2	Sample 3
Positive Model	Buy	Buy	Neutral	Neutral
Negative Model	Neutral	Short	Neutral	Short
Final Prediction	Buy	Neutral	Neutral	Short

For a symbolic word feature set, NBSVM is applied [Wang & Manning 2012]. For distributed representations, the following models are considered: sentence/paragraph vector model [Le & Mikolov 2014], and a Recurrent Neural Network Language Model [Mikolov *et al.* 2010]. These models are preferred due to the state of the art performance in sentiment classification by a recent ensemble model [Mesnil *et al.* 2014].

The original design for the given systems is a binary classification task. The current work is multi-class classification with three labels. Two solutions are available: convert to a multi-class classification task or use a one-vs-all classification strategy.

A multi-class solution reduces the number of samples of each category in the training set. The one-vs-all method can perform a generative maneuvering where instead of one model we create two, namely, a Positive Model for buy decisions and a Negative Model for shorting decisions. In both models, the opposing class will be a neutral (do nothing) label. Table 4.6 lists all potential cases to combine predictions of the generative models. The risk on returns is reduced by diffusing the predictions of both Buy and Short to Neutral. The text in such a scenario could have high volatility conditions, and therefore, avoided.

Table 4.7: Event Matrix as Input to Time-Series Model - Minute Frequency

	$CVX	$BP	$XOM	$SLB
Timestamp				
08-03-2015-11:30	-	-	-	-
08-03-2015-11:31	E	-	-	-
08-03-2015-11:32	-	-	E	-
08-03-2015-11:33	-	-	-	-
08-03-2015-11:34	-	-	-	-
08-03-2015-11:35	-	E	-	-
08-03-2015-11:36	-	-	-	E
08-03-2015-11:37	-	E	-	-
08-03-2015-11:38	-	-	-	-
08-03-2015-11:39	-	-	-	E

4.5 Event Matrix

An Event Matrix is designed to align the timestamp of an event with its stock symbol. Event matrices are utilized by both the time-series model and for trading metrics. Table 4.7 uses a made up example to best describe an Event Matrix for a time-series model. The "E" inside the matrix identifies a clustered tweet event. The time-series model will use the information to know when and what stock to examine.

One tweak is needed for trading metrics. The values inside the matrix now correspond to output labels: 1 is POSITIVE, 0 is NEUTRAL, -1 is NEGATIVE. The Event Matrix in Table 4.8 can be used to make buy/sell trading decisions. The hypothetical trades in the example are as following: buy CVX at 11:31, short XOM at 11:32, and short SLB at 11:39.

Table 4.8: Event Matrix as Input for Trading - Minute Frequency

	$CVX	$BP	$XOM	$SLB
Timestamp				
08-03-2015-11:30	-	-	-	-
08-03-2015-11:31	1	-	-	-
08-03-2015-11:32	-	-	-1	-
08-03-2015-11:33	-	-	-	-
08-03-2015-11:34	-	-	-	-
08-03-2015-11:35	-	0	-	-
08-03-2015-11:36	-	-	-	0
08-03-2015-11:37	-	0	-	-
08-03-2015-11:38	-	-	-	-
08-03-2015-11:39	-	-	-	-1

4.6 Model Type: Time-Series Models

Past patterns capturing price reactions to headlines may be representative of future signals. While technical analysis traders are continuously looking at signals, Event Analysis is confined to a much smaller subspace. Simply put, the attempt is only to capture trading behavior in response to the news. With this in mind, we would like to represent large positive, large negative, and neutral reactions.

As mentioned previously, the input for time-series Models is a pre-event region of 30-minutes. Thus, regression systems use past 30-minutes of price movements to predict future shifts. We convert the price change prediction to trading logic. For example, if a Linear Regression system predicts a price increase of 50 cents, then buy if the buy criterion is any move larger than 30 cents. Essentially, while the learning algorithm is designed for regression, the output is classification.

Classical techniques such as Linear Regression, Support Vector Regression, and

Kernel Ridge Regression use default parameters and are implemented using Sci-Kit Learn [Pedregosa *et al.* 2011]. Neural Networks, however, have infinite possibilities for architectures. To stay within reason, we have focused on one Feedforward Neural Network and three Recurrent Neural Networks. The reason for multiple structures is to compare the Gated Recurrent Unit (GRU) versus the Long Short-Term Memory (LSTM) Unit.

Model names for neural networks are labeled in a way to highlight architecture. The feedforward neural network, FWD 0 RNN 6 Dense, has zero RNN layers and six fully-connected hidden layers. RNN 1 GRU 1 Dense has one GRU layer and one fully-connected hidden layer. RNN 4 LSTM 1 Dense has four LSTM layers and one fully-connected hidden layer. And lastly, RNN 7 LSTM 0 Dense, has seven LSTM layers and zero fully-connected hidden layers. Every neural network has a single neuron as its output layer. The output layer does not have an activation function since the neural networks learn to predict values for price shifts. Training enhances these predictions by minimizing the mean squared error function. Once complete, the projection is converted to class labels using BUY/SHORT criteria. A resource that we used for our model designs is presented in [Goodfellow *et al.* 2016].

The only non-regression method, K-Nearest Neighbors, aims to gather clusters using class definitions. During training, a labeled Event Matrix is passed to the model. Once an event is identified, historical data for that company are retrieved and used as a query signal. The query is compared against all time-series stored in class clusters. A similarity is checked between the query and representative signals using DTW. Every representative signal has a weighted vote. Weights are subject to change during training. Majority

voting determines the final class. It is worth noting that no conclusive evidence is given in favor of a particular stock representation. Therefore, the study compares time-series on both price shifts and percentage returns.

A visual example is presented to further describe the process. However, the illustration is not associated with a real Twitter event. Figure 4.1 highlights three identified signals stored in their respective cluster (CCE - Positive, SWX - Neutral, WWC - Negative). Additionally, Figure 4.1 shows historical data of a query signal.

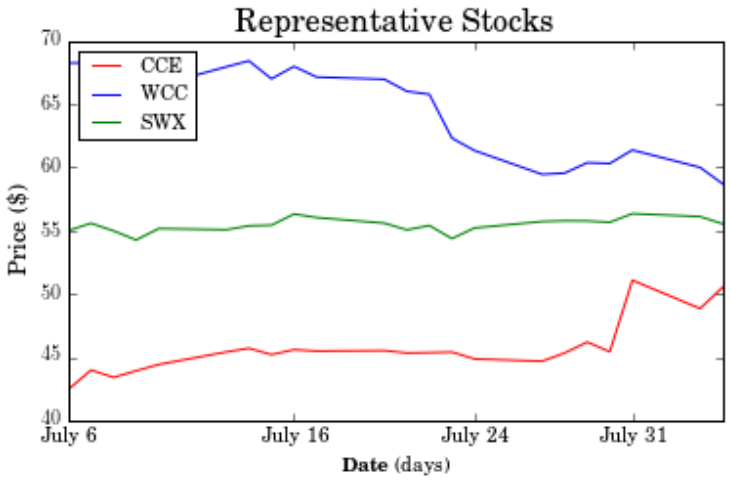


Figure 4.1: Representative Time-Series Signals

The query signal is evaluated against the representative signals to measure similarity. Table 4.9 displays the DTW distance calculated. It can be seen that similarity is closest to the Positive class. Unlike the example mentioned, the time horizon is pre-fixed only to include data before an event. Furthermore, each category has many representative signals within its clusters. Based on the overall strategy, the system attempts to answer the question "*Is the stock primed for a large movement?*".

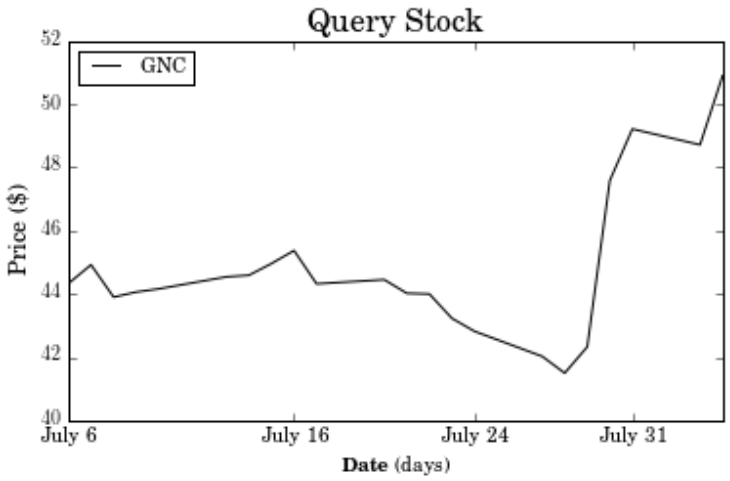


Figure 4.2: Query Signal

Table 4.9: Calculated DTW Distance - Query/Representative

Similarity between GNC and CCE	3.07
Similarity between GNC and WWC	7.06
Similarity between GNC and SWX	4.24

Chapter 5
Experimental Results

5.1 Analysis of Twitter Datasets

Before profit results, we analyze stored collections and highlight some helpful techniques found to reduce useless events. A filter is necessary because Twitter is full of casual conversations.

The event clustering algorithm moves towards using Twitter in a manner similar to those of news headlines. We begin by not including retweets, and in doing so, all saved tweets are unique and independently produced. The number of retweets is unknown because the storing routine rejects them immediately.

Using the clustering method, the total number of original tweets stored for the energy and company-centric collections are 85,910,846 and 1,557,811, respectively. Since language has grammatical peculiarities, the focus is only on English tweets.

Removing retweets aids event clustering by avoiding the arrival of identical texts. Unfortunately, a similar concern arises given the pervasiveness of automated bots on Twitter. Luckily, the Twitter API provides an origination of source for each tweet. A source tag registers if a tweet was formed using an iPhone app, Blackberry app, any third-party app, or the Twitter website itself. By examining sources, a filter is built to block automated tweets.

In viewing preliminary events, few sources appear to dominate as candidates for

Table 5.1: Statistics on Stored Tweets for Energy Collection

Tweet Collection	Tweet Count	% of Collection
Total	85,910,846	100%
English	63,223,739	73.59%
Not Blacklisted	42,873,892	49.91%

Table 5.2: Statistics on Stored Tweets for Company-Centric Collection

Tweet Collection	Tweet Count	% of Collection
Total	1,557,811	100%
English	1,398,981	89.80%
Not Blacklisted	877,054	56.30%

presenting repetitive texts. For example, http://ifttt.com is a third-party provider that enables users automation for generating tweets anytime a self-defined outcome occurs. Using this third party provider, every morning, one can automatically tweet the weather forecast. If multiple users have this automation, then identical tweets will be generated around the same time. A blacklist containing sources that exhibit such behavior is designed to mitigate the problem. The list is by no means exhaustive; however, it is an essential step for the event clustering algorithm. Either blacklisting sources or accepting trusted sources are our recommendations to move towards identifying noteworthy news on Twitter.

Tables 5.1 and 5.2 state figures for filtered documents; English Filter and Blacklisted Source Filter. Approximately, half of the energy tweets are filtered from grouping into events. These high percentages show the influence that bots have on Twitter.

Table 5.3: Comparitive Statistics on Event Collections

Collection	Total Events	Training Size	Validation Size	Testing Size
Energy	81,466	36,553	21,389	23,524
Company-Centric	35,453	19,788	7,352	8,313

5.2 Analysis of Event Collections

After applying the event clustering algorithm on filtered documents, this section presents the facts and figures between collections. The numbers change as the exit strategy parameter is modified. Thus, a 60-minute exit strategy is used to maintain consistency across datasets. An examination of which exit strategy works best is presented later.

Table 5.3 reveals counts on the number of events created using time-periods from a previous section. As mentioned, large percentages of events are prevalent in the validation and testing sets. This stays consistent in ensuring fairness to the concern of overfitting in finance.

The events created are used similarily to news headlines. Each one is associated with a converted tradeable market time. The conversion is essential since it either allows or disallows the ability to trade certain Twitter events. For example, if a cluster is identified outside of trading hours, we can either ignore it, or trade it at market open. Figures 5.1 and 5.2 shows a percentage comparison of all events between both datasets seperated by hour. In both scenarios, a large proportion of clustered events, 81.7% and 69% are at market open. Thus, the decision is to use events at market open. There are no events after 3:00 pm EST, this is because we are using a 60-minute holding period. All events that need to hold the stock for a longer period of time will be a subset of either

collection. Similarily, events at market open are not available for time-series models since they require a 30-minute pre-event region.

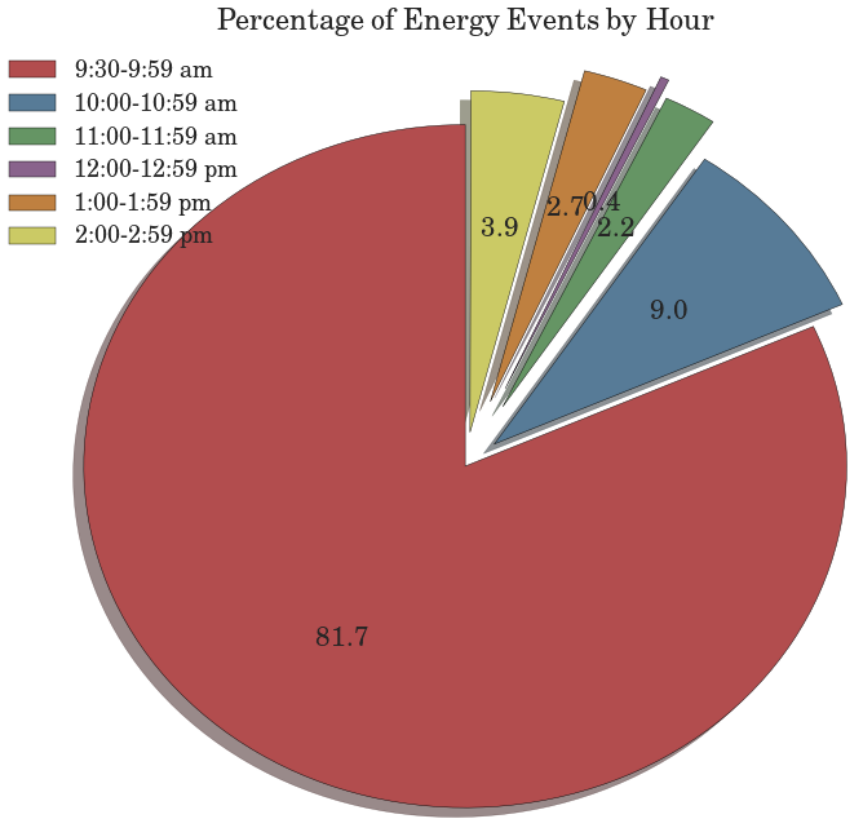


Figure 5.1: Energy Events by Hour

Similar logic applies to trading days. The strategy is to ignore all evening events on Friday, while keeping all weekend (Saturday and Sunday) tweets. Including weekends increases the number of events on Mondays; this is seen in Figures 5.3 and 5.4. The decisions described here are the same choices traders have to consider when incorporating

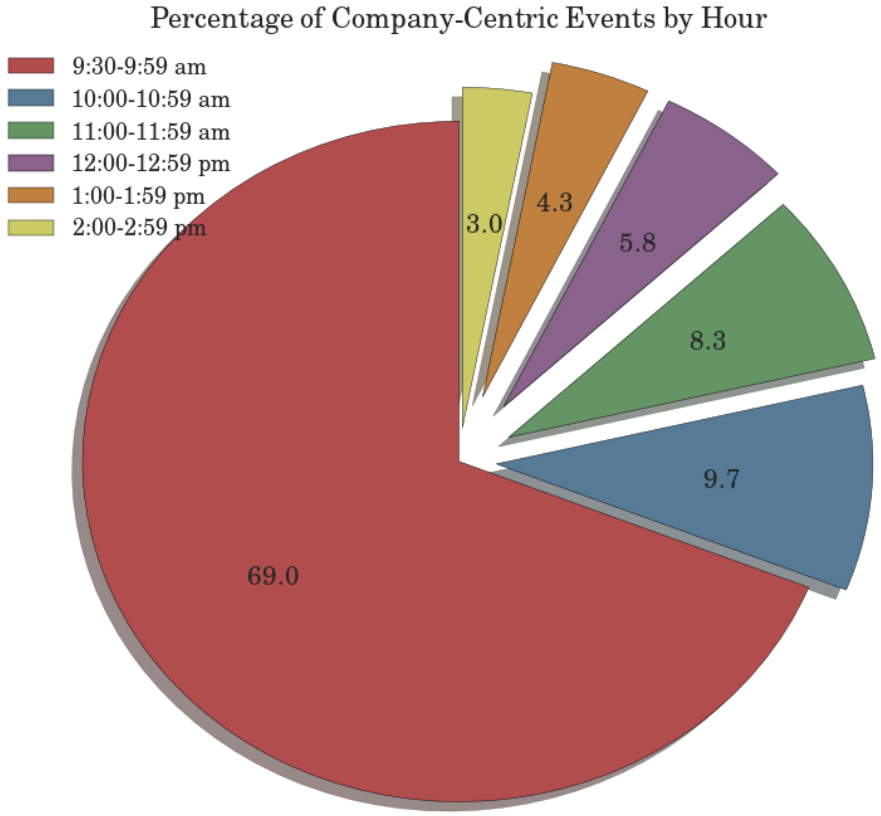


Figure 5.2: Company-Centric Events by Hour

morning and weekend news. The effect of news from outside of trading hours may explain why the largest price movements in a trading day happen at market open.

A significant number of events have been merged and classified as lagged-events. Each event is saved for 4 hours. If any new event has a high similarity to a past event, it is considered lagged. Merged totals are 169,373 and 32,258 events, respectively, for energy and company-centric collections. Thus, 47.6% of company-centric and 67.5% of energy

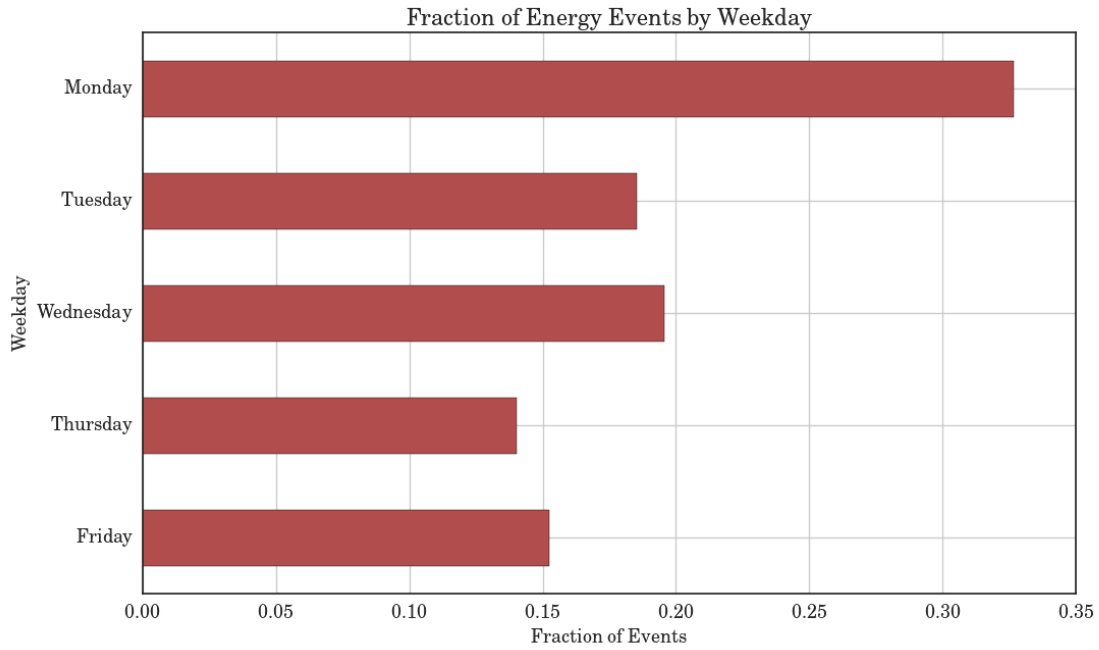


Figure 5.3: Fraction of Energy Events by Weekday

events are merged.

With additional information provided by the Twitter API, analysis can go beyond just the text of tweets. Hashtags are used as topical indicators. Figures 5.5 and 5.6 list the 10 most frequent hashtags across collections. Hashtags give a quick look at top categories clustered by the algorithm. The top ten company-centric hashtags are all relevant. The same is not true for energy events. The "#nowplaying" hashtag does not fit an energy trading context. Third party applications cause the "#nowplaying" hashtag to form as an event. For instance, this third-party app allows users to tweet currently playing songs. Ordinarily, music events should not appear for a financial purpose, however, the keyword "energy" also fits into a music context. A posthoc analysis can find a source for blacklisting. However, these are inevitable, so the learning algorithms are left to figure

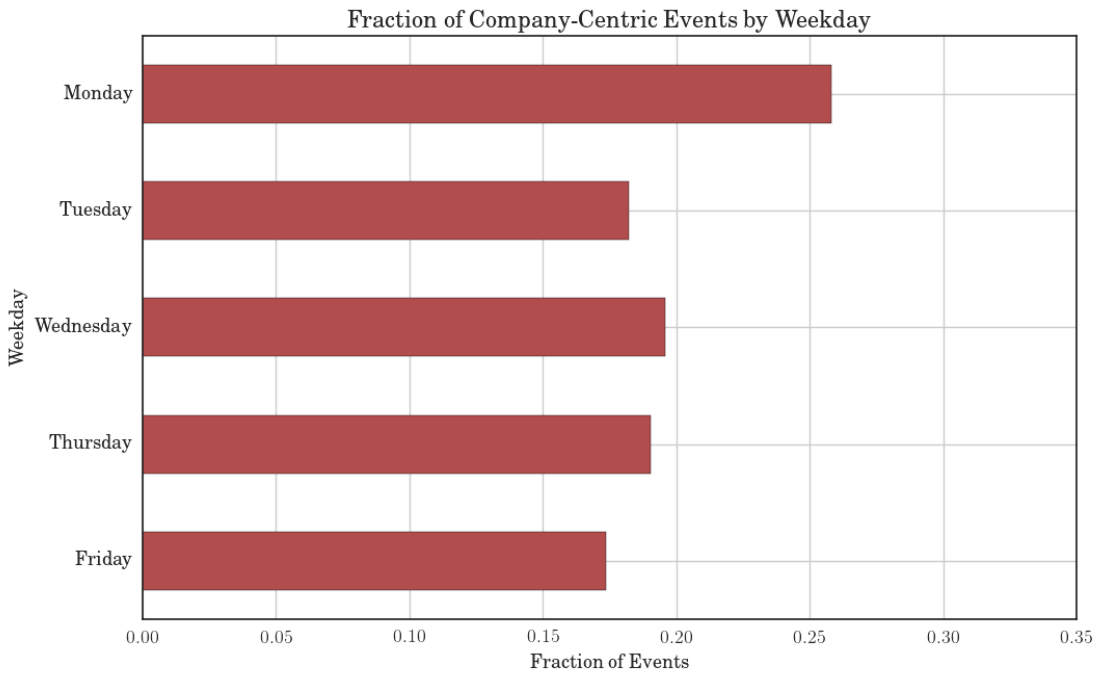


Figure 5.4: Fraction of Company-Centric Events by weekday

out how to filter out-of-context events. The only other hashtag that seems out of place is "#cop21". This is not the case because "#cop21" represents a United Nations Climate Change Conference.

Instead of blacklisting sources, researchers may prefer whitelisting. Here is a list of a few standouts that stay on topic for financial markets; `http://seekingalpha.com`, `http://stocktwits.com`, `http://www.stocknewswires.com`, `http://www.estimize.com`. Top 20 sources are provided in a full list form in Tables A.4 and A.5.

While not taken advantage of in this work, it may be possible to examine which users produce relevant events. A listing of User ID's with the highest frequency of tweets is provided in Tables A.6 and A.7.

Lastly, while energy events use a specified ETF for determining price movements,

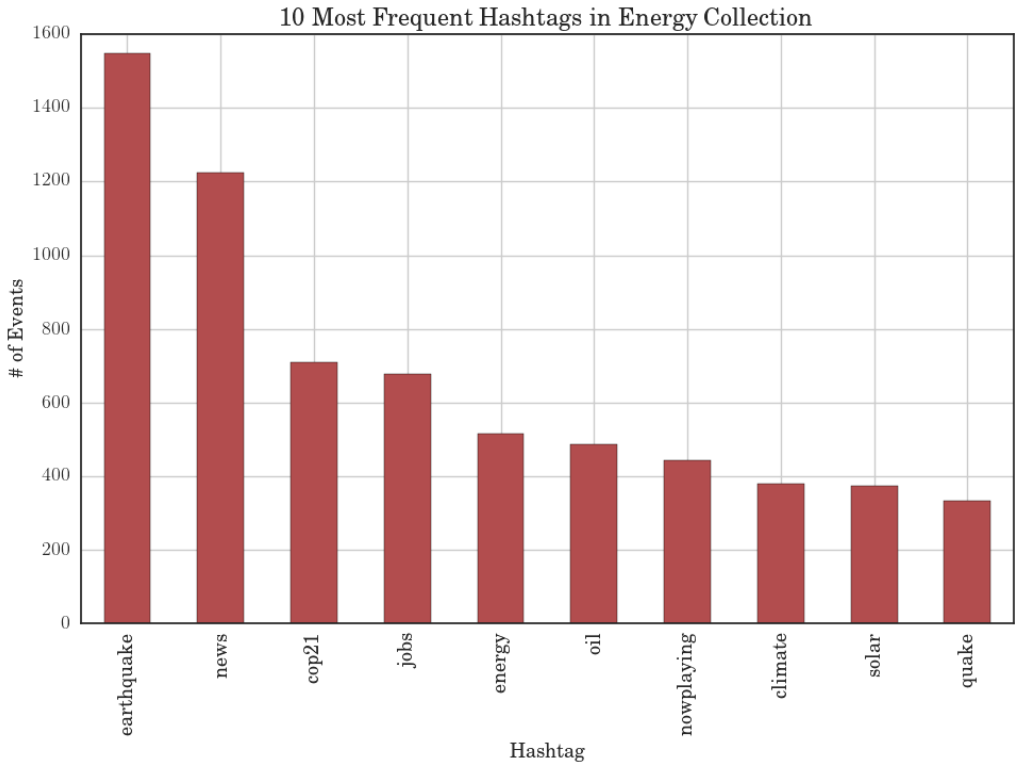


Figure 5.5: Top 10 Hashtags in Energy Events Collection

company-centric events depend on users' tweets. Figure 5.7 list the number of events for the top 20 stocks mentioned.

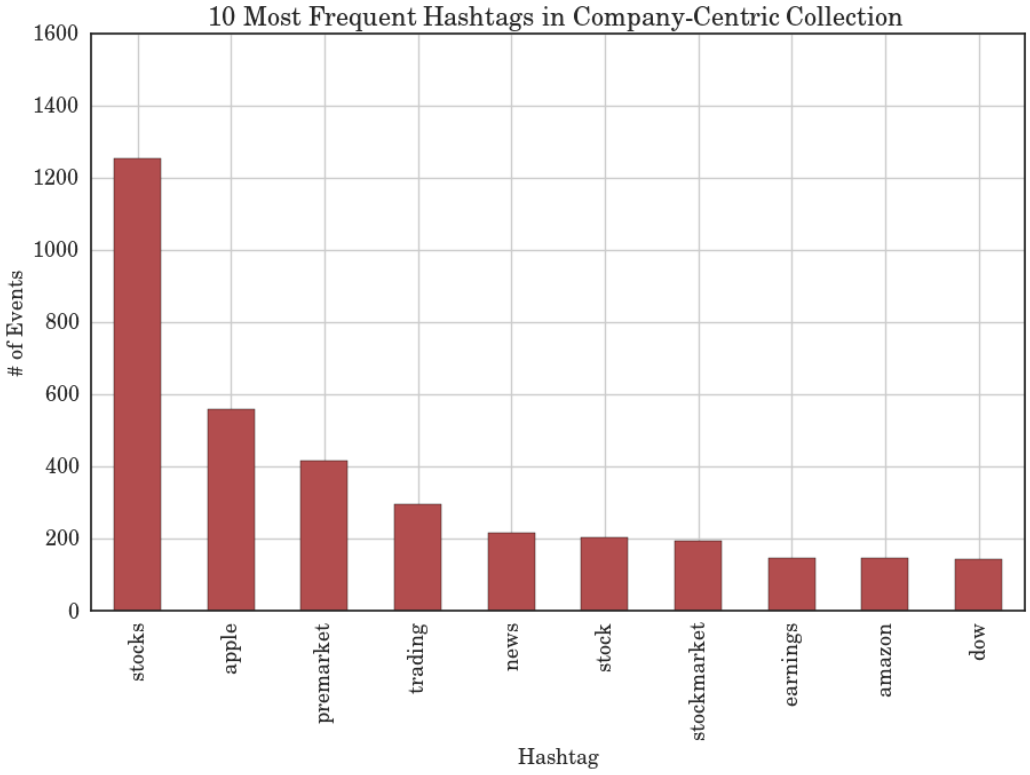


Figure 5.6: Top 10 Hashtags in Company-Centric Events Collection

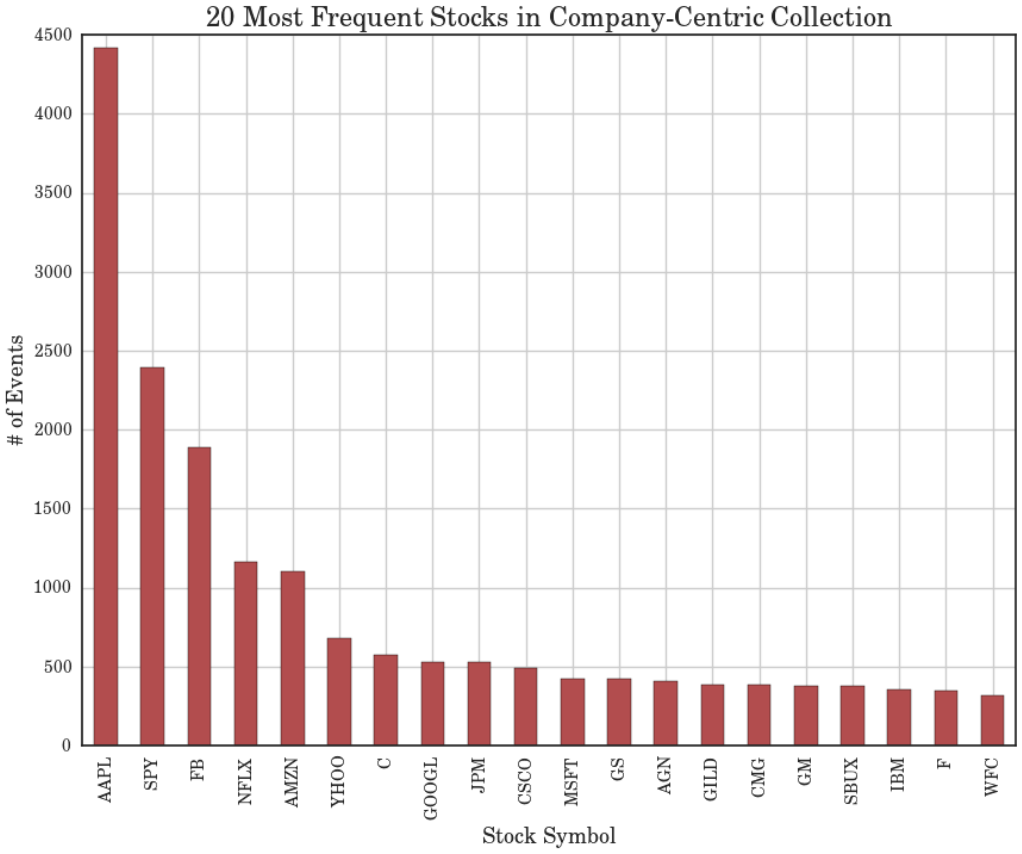


Figure 5.7: Top 20 Stocks in Company-Centric Collection

5.3 Performance of Event Collections

Keeping consistent with a 60-minute holding strategy, this section compares unannualized Sharpe Ratios across collections. Annualizing is saved for later as special care is needed. The energy collection uses an ETF called XLE (Energy Select Sector SPDR ETF) for price movements.

Annualizing Sharpe Ratios is a matter of scaling. If we have monthly returns, then we would multiply the Sharpe Ratio by the square root of 12. Daily returns would scale by the square root of 252 and not 365; since there are 252 trading days in a year. In using a one-hour holding period, we may be tempted to multiply 252 by the number of hours in a trading day. However, this should not be the case because the strategies do not provide opportunities to trade every single hour. Instead, the learning algorithms do present enough events to achieve these returns at least once a day. Thus, we scale returns daily to annual rather than hourly to annual.

To allow scaling of Sharpe Ratios, an underlying assumption is that the returns are independent and identically distributed. A study suggests an overestimation of Sharpe Ratios when the assumption does not hold [Lo 2002]. Overstated Sharpe Ratios stem from the presence of serial correlation in returns [Lo 2002].

The energy collection has many events that arrive within minutes of each other. The returns on such events will be serially correlated. For this purpose, our comparisons in Figure 5.8 and 5.9 do not scale Sharpe Ratios. The sign of Sharpe Ratios does not change with scaling; so it will still indicate potential profits or losses.

The company-centric collection has independent events spread out across many

financial securities. There is less likelihood for serial correlation in returns. When considering only company-centric results, Sharpe Ratios will be annualized.

Figure 5.8 and 5.9 highlight a substantial difference in the potential for profitability between the two collections.

For energy tweets, there are many inconsistencies in Sharpe Ratio across the validation and testing set. Recall that both sets are unseen samples for learning algorithms. So any significant disparity between sets shows that the models do not perform well in changing market conditions.

On the other hand, the company-centric collection does well and is mostly consistent. Two of the most promising models are language models; NBSVM and PARAGRAPH. The decisions used from these systems would present meaningful gains for day traders over the two-month period. Sharpe Ratios give a general feeling of a trading strategies profitability. Specifics on the profit per share and the distribution of returns are presented in a later section.

While company-specific events seem encouraging, it is tough to predict price movements on general events. One reason why is because of how events are auto-labeled. Since clusters arrive at a high velocity, using one ETF as a price reference will tangle the influence of the initial story. When an event causes the ETF to jump, many clusters arriving at the same time will subsequently be labeled as positive. Given the scenario, the potential for language models is diminished by adding confusion to the training set.

An issue occurrs when multiple stock symbols show up in a tweet. If "$AAPL" and "$FB" are listed, then we must choose to accept, either, both symbols, only one or none.

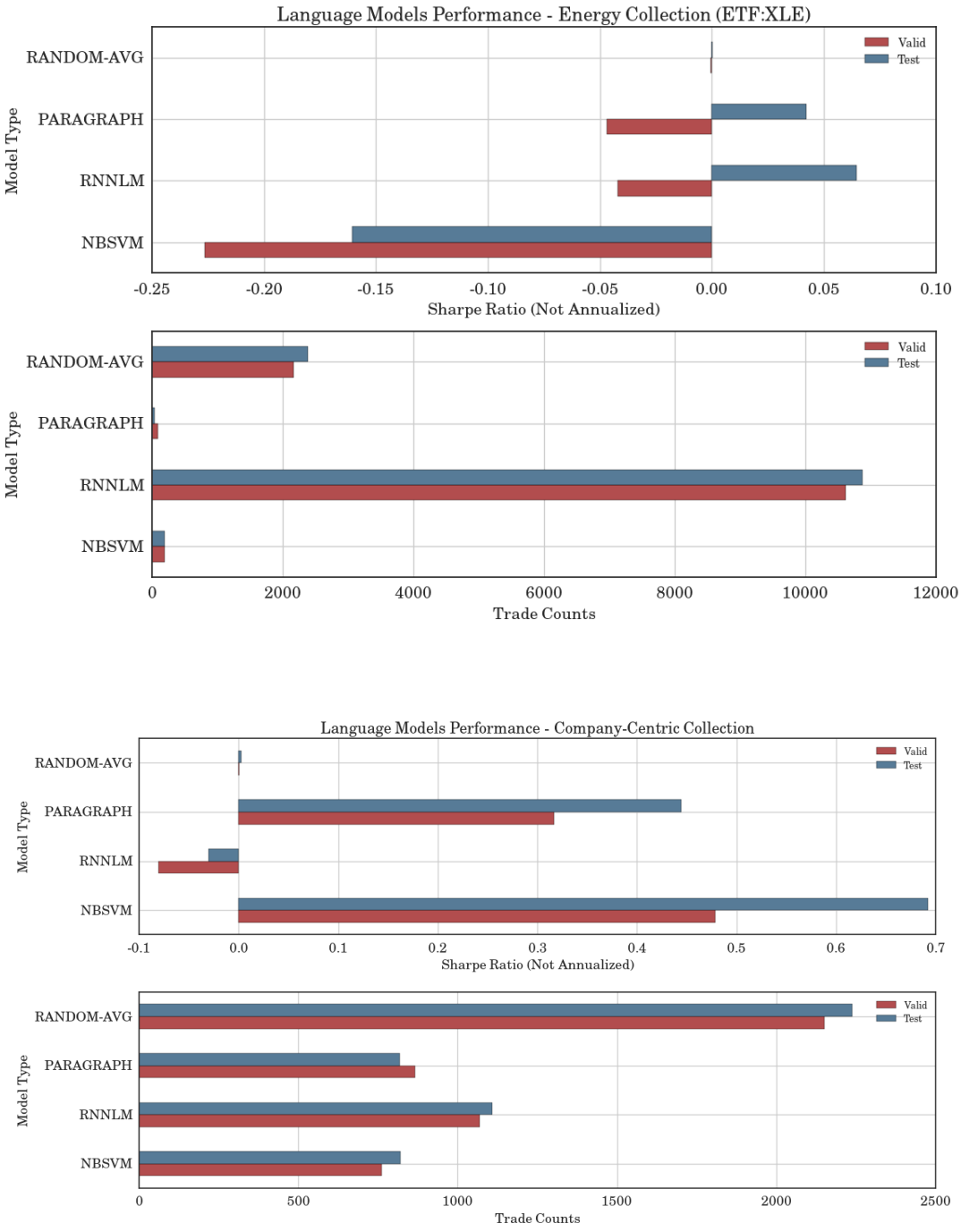


Figure 5.8: Comparison of Collection Performance on Language Models

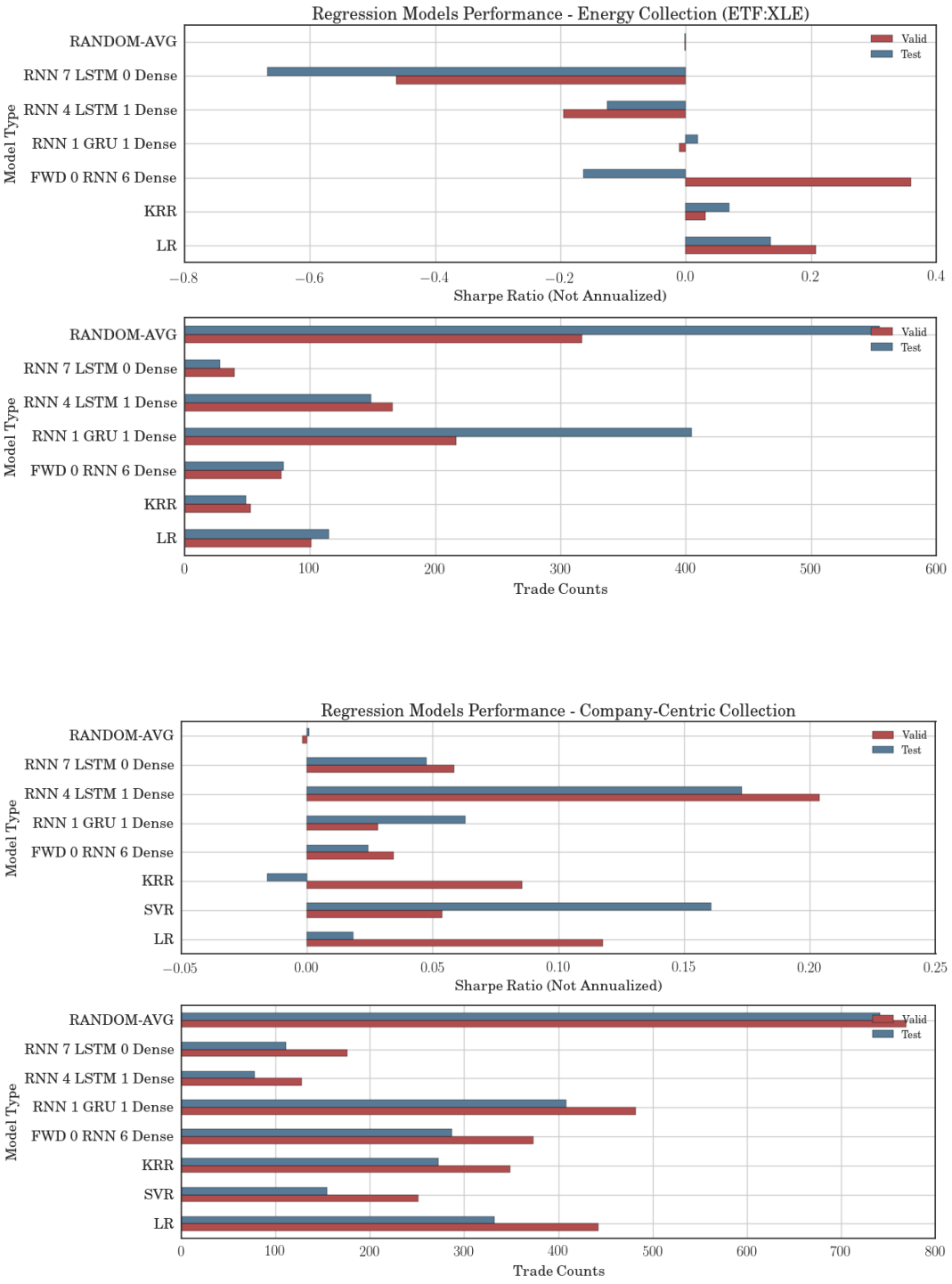


Figure 5.9: Comparison of Collection Performance on Regression Models

Multiple events with identical text are created if all symbols are used. Initial findings showed that doing this negatively affected the Sharpe Ratio. Thus, as an initial solution we only choose the first symbol alphabetically. Unfortunately, a similar solution for energy events did not seem to exist.

5.4 Exit Strategy for Language Models

By outperforming the energy collection, company-centric events are the clear choice for further analysis. Thus, all results going forward will be for this collection.

A tough decision for traders is to figure out when to exit a trade. The experiments measure performance on intraday holding periods of 30, 60, 120, and 240-minutes. Figures 5.10, 5.11, 5.12, and 5.13 show Sharpe Ratios and the number of trades for each language model. Figures include ensembles that form by using the most common prediction from individual models. Given the doubt of serial correlation in the company-centric collection, a conversion is made to an Annualized Sharpe Ratio.

Aside from the poor performance of the RNNLM model, it is clear that all models perform remarkably well. With mostly all Sharpe Ratios near or above 10, the most consistent exit strategy is a holding period of 120-minutes. With monthly or yearly holding periods, a Sharpe Ratio of greater than three is considered excellent. Shorter-term and high-frequency traders prefer higher Sharpe Ratios. Due to smaller returns, a larger amount of capital is needed to invest in high-frequency positions. Since a significant capital investment is required, lower risk opportunities are adequate.

Trade counts, greater than or close to 500, boost confidence in the results as it ensures

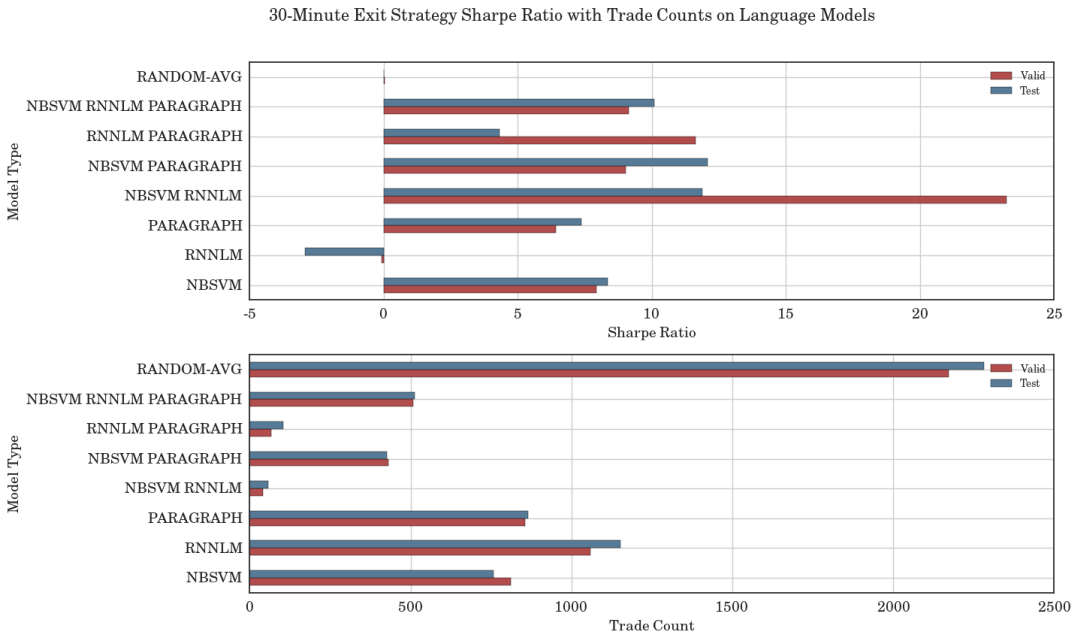


Figure 5.10: Performance of Language Models on 30-Minute Exit Strategy

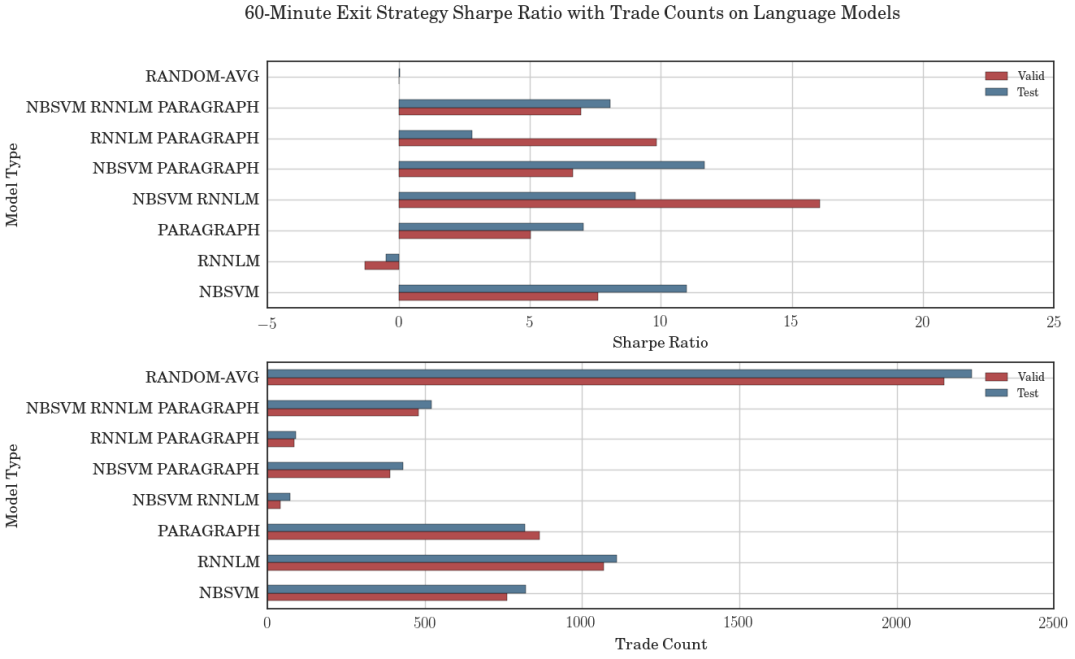


Figure 5.11: Performance of Language Models on 60-Minute Exit Strategy

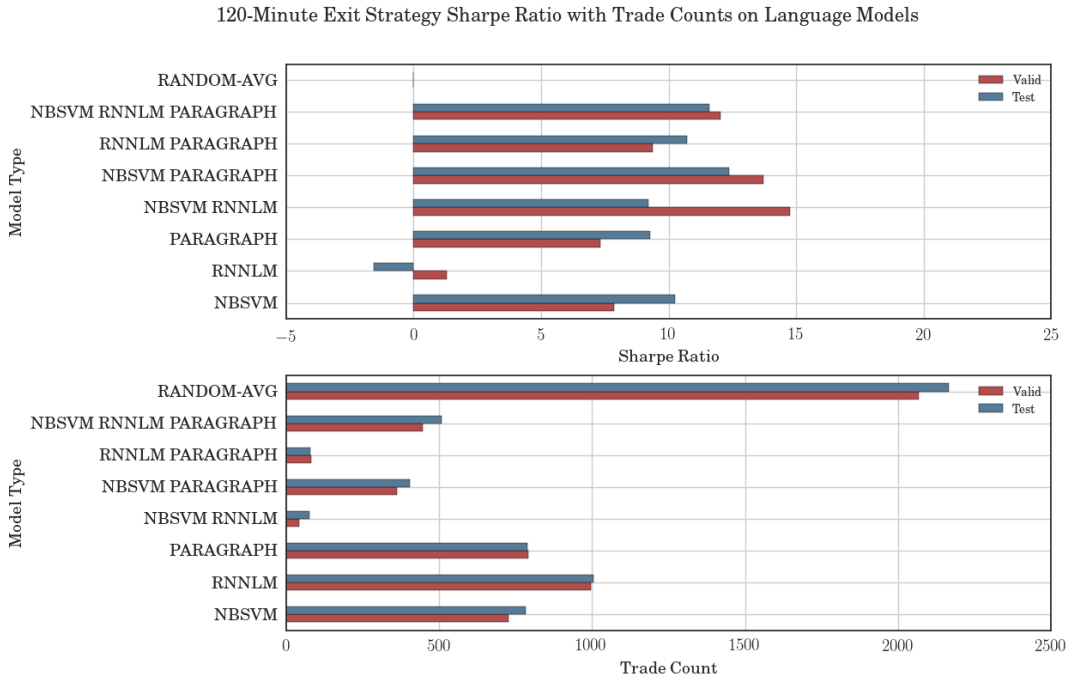


Figure 5.12: Performance of Language Models on 120-Minute Exit Strategy

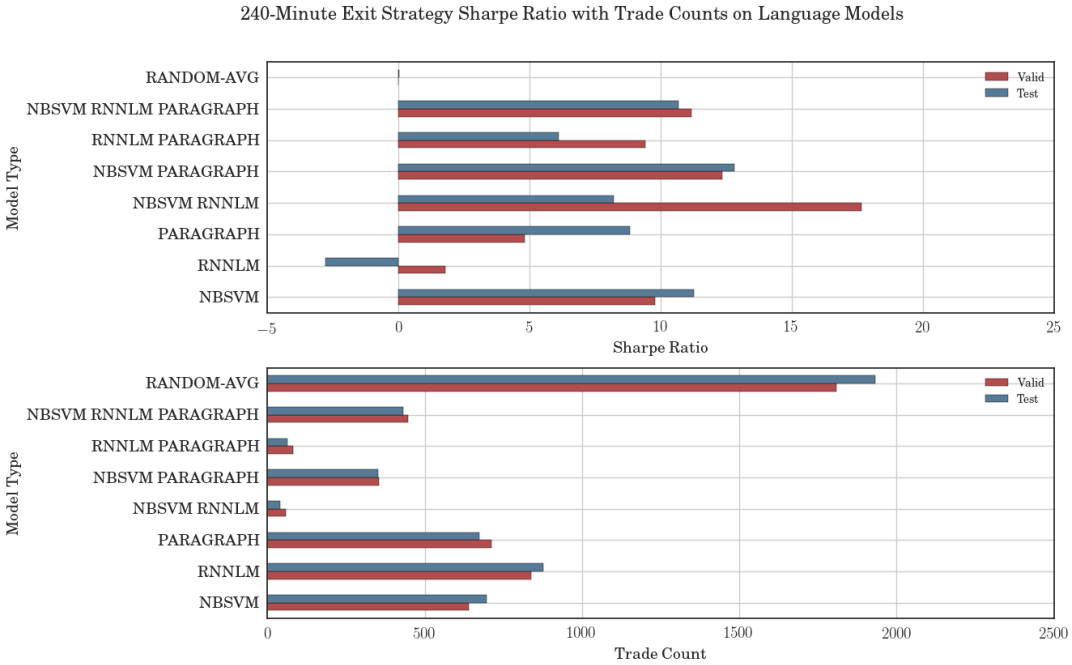


Figure 5.13: Performance of Language Models on 240-Minute Exit Strategy

no single outcome dominates averages. The only ensembles that have a small number of trades are pairs that include RNNLM; this shows that there is a limited union of decisions between RNNLM and other language models. The displayed results fit the requirements of short-term traders, noted above.

5.5 Exit Strategy for Regression Models

Regression performances across exit strategies are displayed in Figures 5.14, 5.15, 5.16, and 5.17. Sharpe Ratios ranging from -2 to 3 fail to impress when compared to language models. A lower range may be partly due to being evaluated on a small subset of the data that language models trade on.

Regression models forecast price shifts based on the past 30-minutes of movements. A significant portion of cases appear at market open. Therefore, less than 30% have the pre-event region required for projection. Even with a smaller training set, time-series models mostly learn to be profitable.

Time-series predictors, unlike language models, show clear differences amongst exit strategies. Longer time periods of 120 and 240-minutes show negative Sharpe Ratios and inconsistencies between the validation and testing set. The 60-minute holding period seems like the best choice for regression models.

It is reasonable to expect shorter term forecasts to be more accurate. On good decisions, allowing a trade more time aids the ability to accumulate greater returns. Since Sharpe Ratios are a function of both risk and returns, a good balance helps. The performance of the 60-minute exit strategy over the 30-minute exit strategy may follow

from this reasoning.

When comparing models, there is not one that outperforms all others across exit strategies. A closer examination of the performance of regression models is required. Both language models and regression models perform well on a 60-minute holding period. For this reason, we continue beyond Sharpe Ratios and later highlight financial metrics using the 60-minute exit strategy.

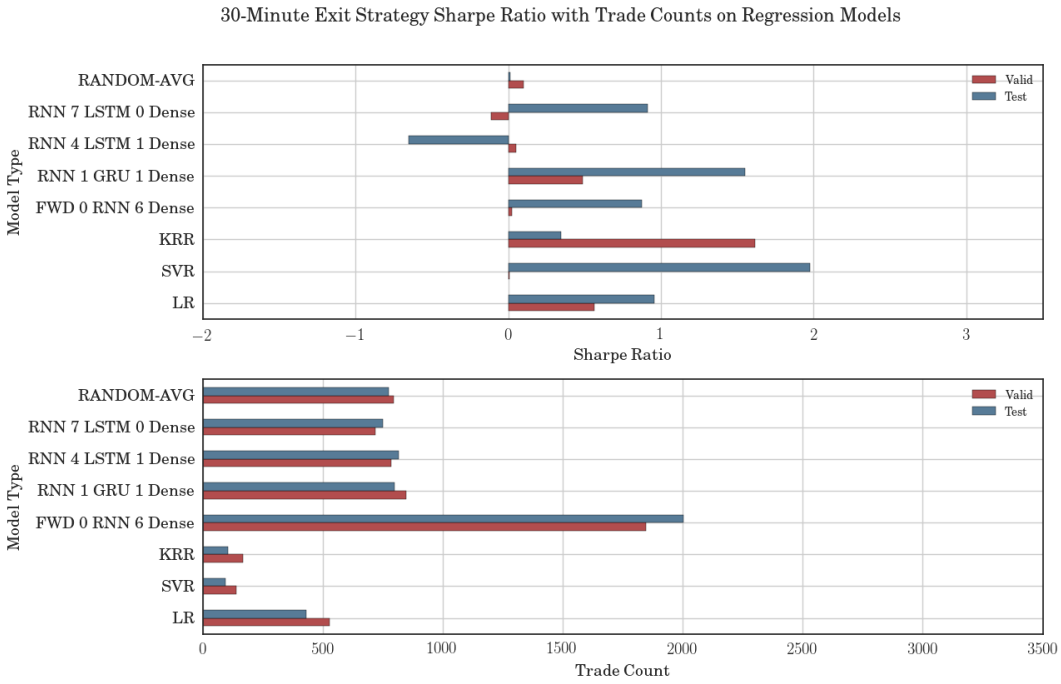


Figure 5.14: Performance of Regression Models on 30-Minute Exit Strategy

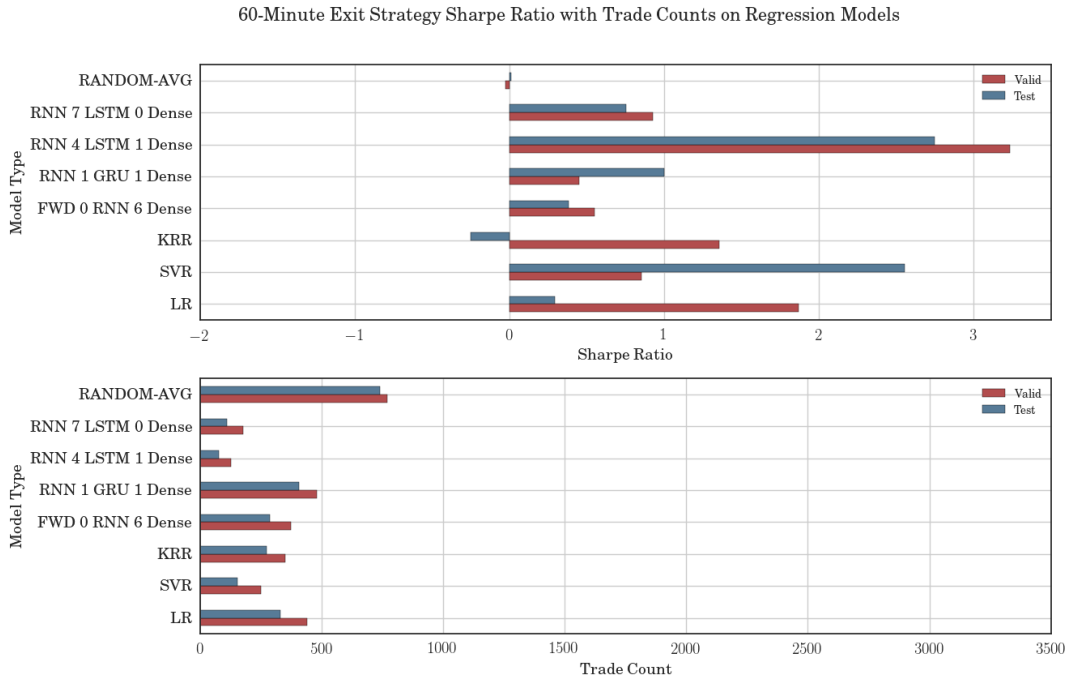


Figure 5.15: Performance of Regression Models on 60-Minute Exit Strategy

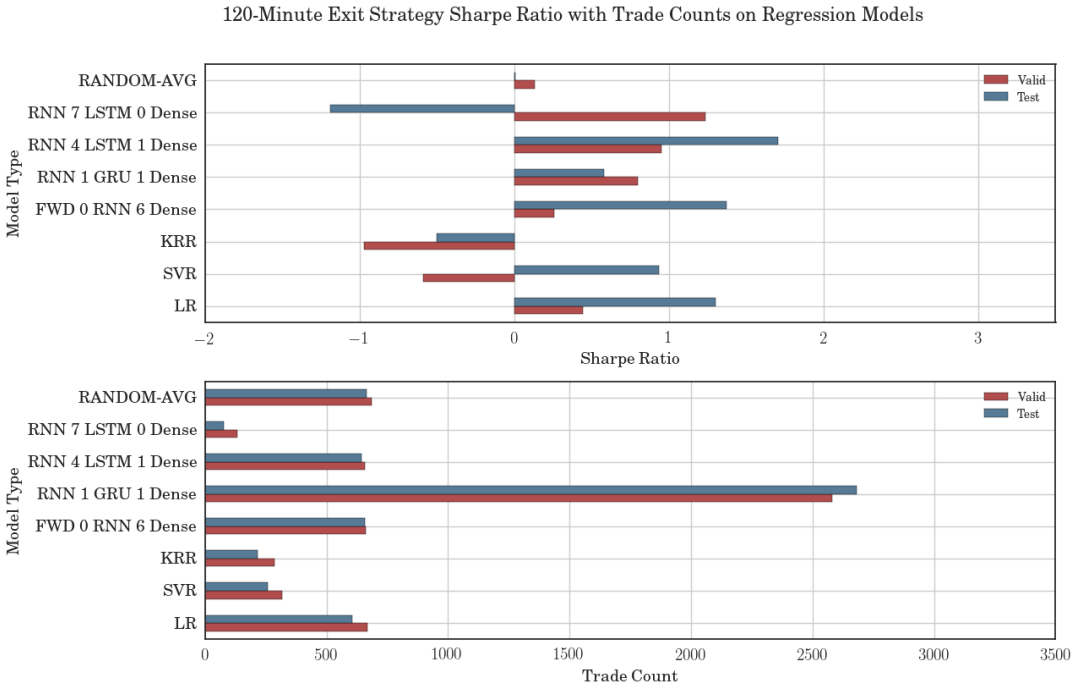


Figure 5.16: Performance of Regression Models on 120-Minute Exit Strategy

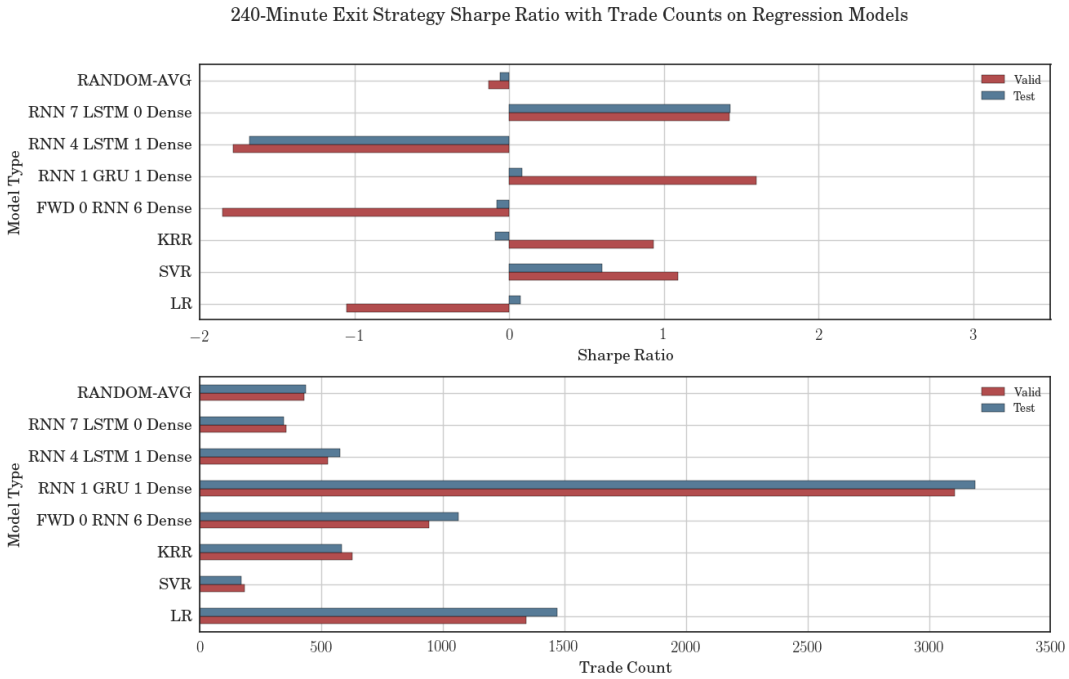


Figure 5.17: Performance of Regression Models on 240-Minute Exit Strategy

5.6 Evaluation Metrics

A Sharpe Ratio acts as a sound indication to accept or reject a trading strategy. Still, a closer inspection of additional metrics is advised. This section considers the distribution of stock returns, profits per share, as well as machine learning metrics of precision, recall, and accuracy.

Figure 5.18 helps in understanding the distribution of percentage returns; presented is a box plot of returns for language models. A box plot highlights the mean, quantiles, and outliers from all trades. The mean and majority of predictions are above 0% for NBSVM and PARAGRAPH. Furthermore, most outliers have large positive gains. Even with high Sharpe Ratios, a detail missed numerically is the evidence that most outliers are positive. Sharpe Ratios are discouraged by risk in either direction. Investors, on the other hand, want notable positive gains. For this reason, some traders prefer to use Sortino Ratios which only adjusts for downside risk. The distribution of returns further persuades investors in using the event clustering algorithm and language models. Comparing the box plot of returns alongside Shape Ratios infers more meaningful interpretations.

By using a histogram in the range of -5% to 5%, we can capture the distribution of where most movements appear. Figures 5.19 and 5.20 show how profitable models should look, while Figure 5.21 displays a model that would lead to losses. There is a natural skew to the right for both NBSVM and PARAGRAPH along with a large number of profitable trades. It is also clear that the majority of returns have a mean close to 0.6%. The average may seem small, but these are accumulated under just 60-minutes and not annualized. The advantage of a shorter holding period is the ability to free up capital.

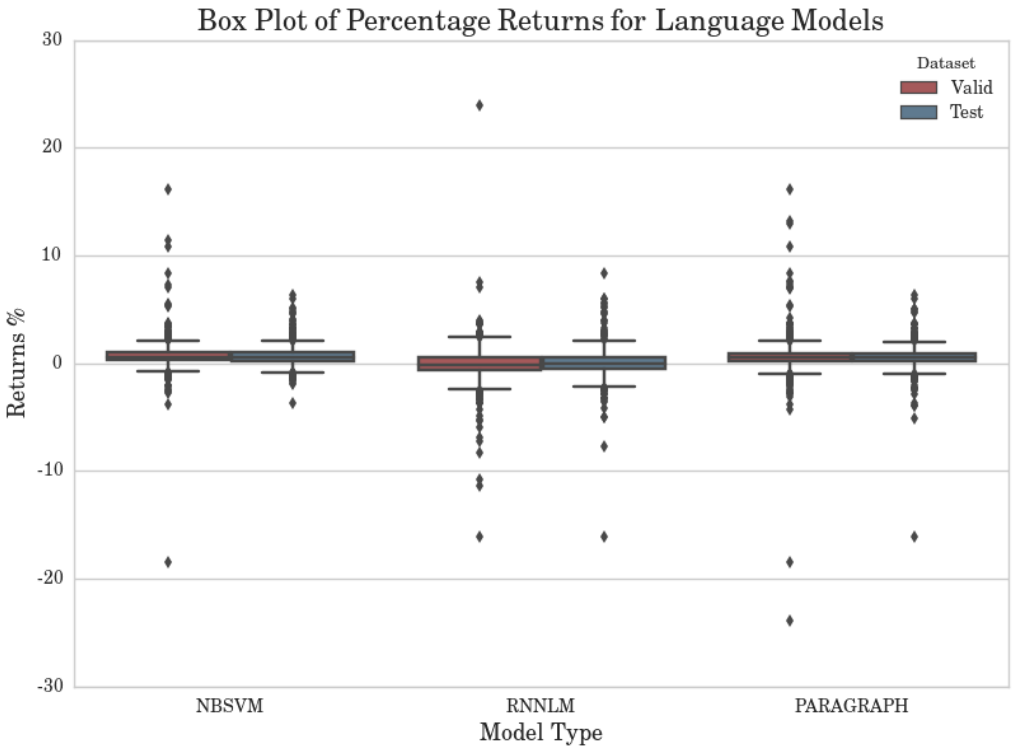


Figure 5.18: Box Plot of Percentage Returns Compared Using a 60-minute Exit Strategy on Language Models

A 0.6% return can be absorbed and reinvested to give compounding effects.

Tables 5.4, 5.5, 5.6, and 5.7 report evaluation metrics for language and regression models on the validation and testing sets. On the evaluation metrics alone, text models undoubtedly outperform time-series models. Still, it is important to compare them seperately since both are trading on different subsets of the data.

Time-series models require an input representation of financial data. Price range varies for stocks, making normalization tricky. Initial inquiries found using returns or log returns, too noisy for learning algorithms. Instead, using price shifts captured in minute frequencies, aid regression models for predictions. Converting the 30-minute pre-event

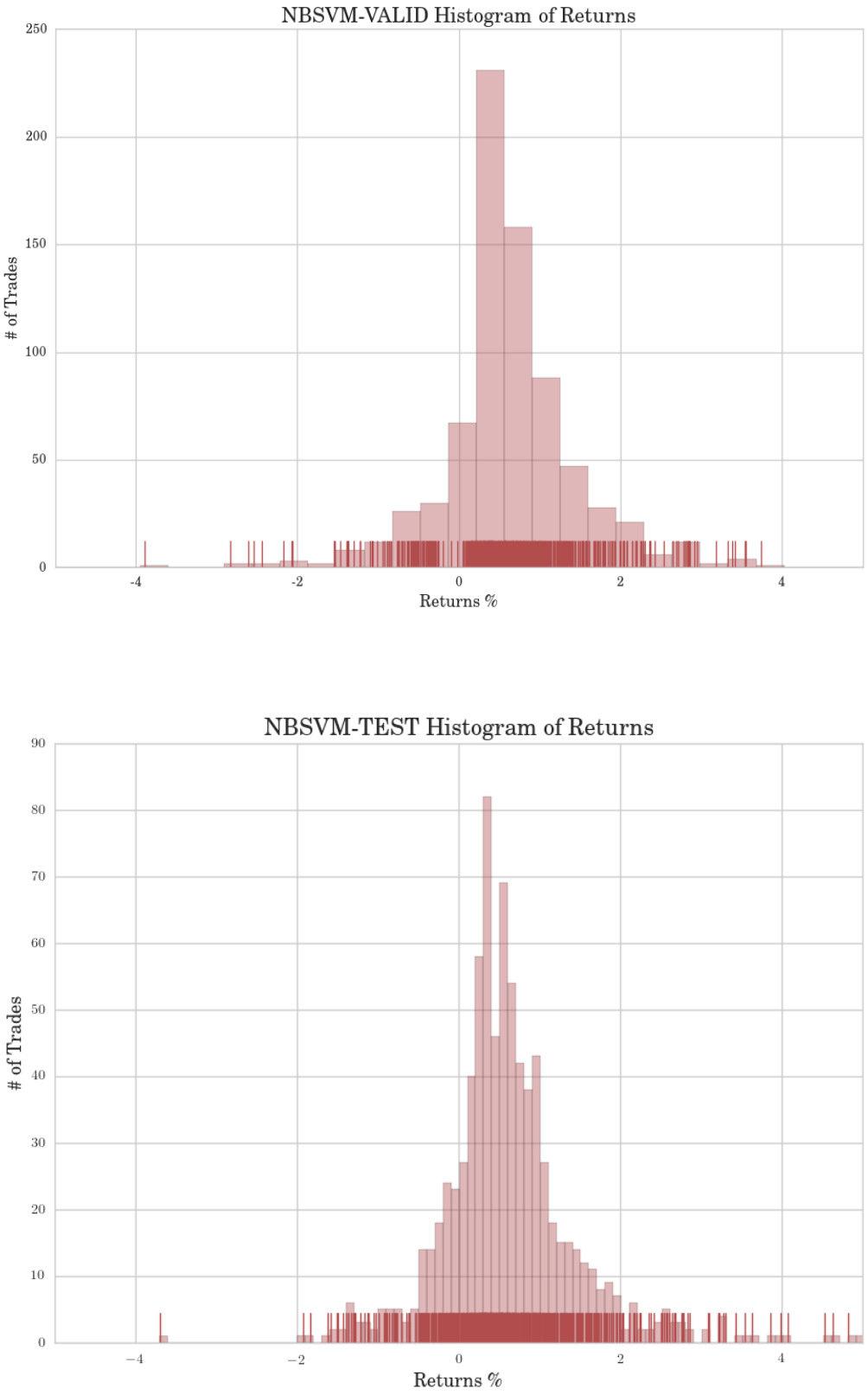
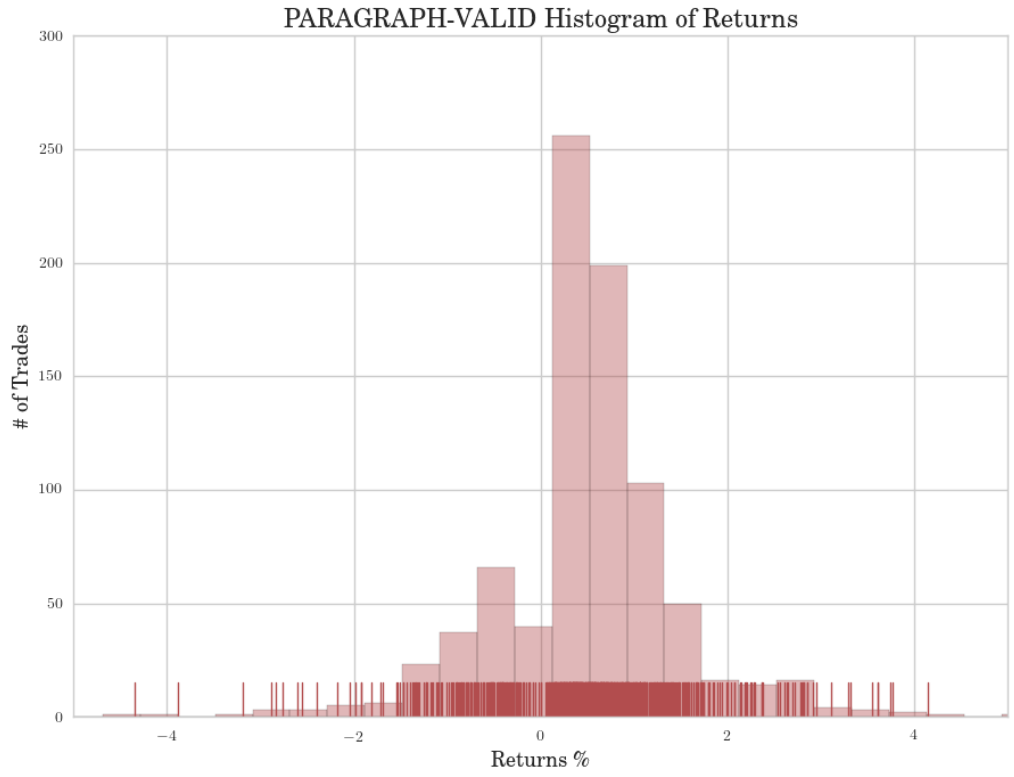


Figure 5.19: NBSVM Histogram of Percentage Returns Focused on Range (-5%,5%)



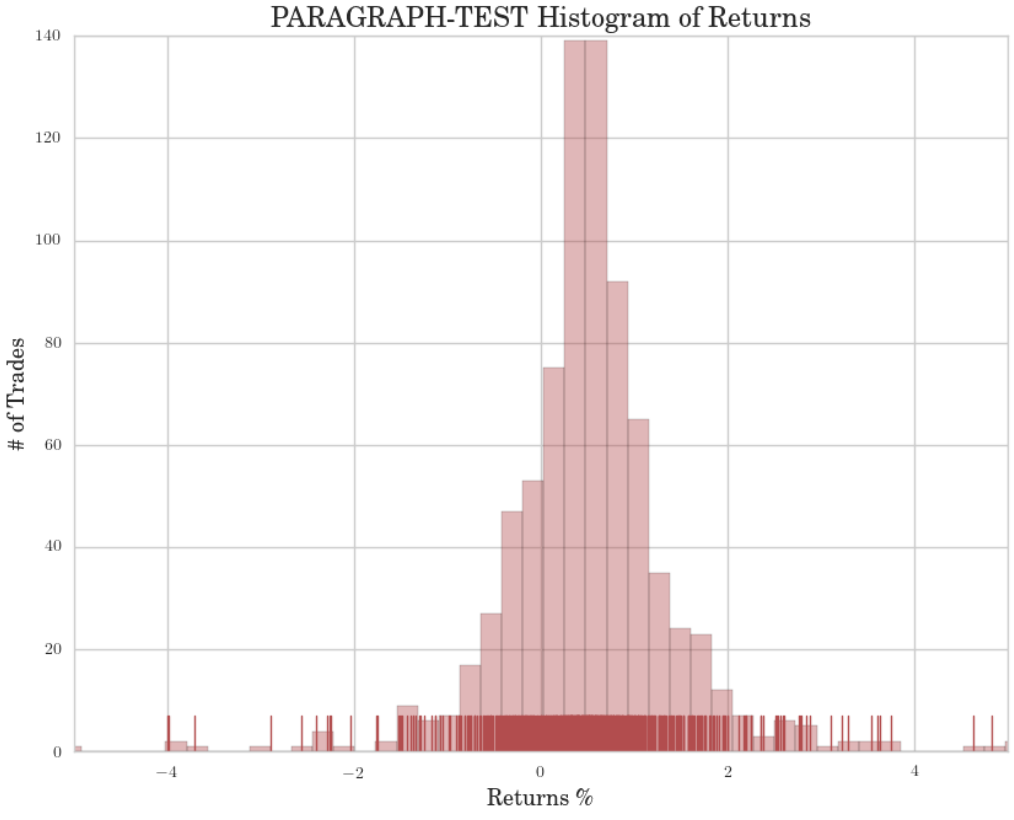


Figure 5.20: PARAGRAPH Histogram of Percentage Returns Focused on Range (-5%,5%)

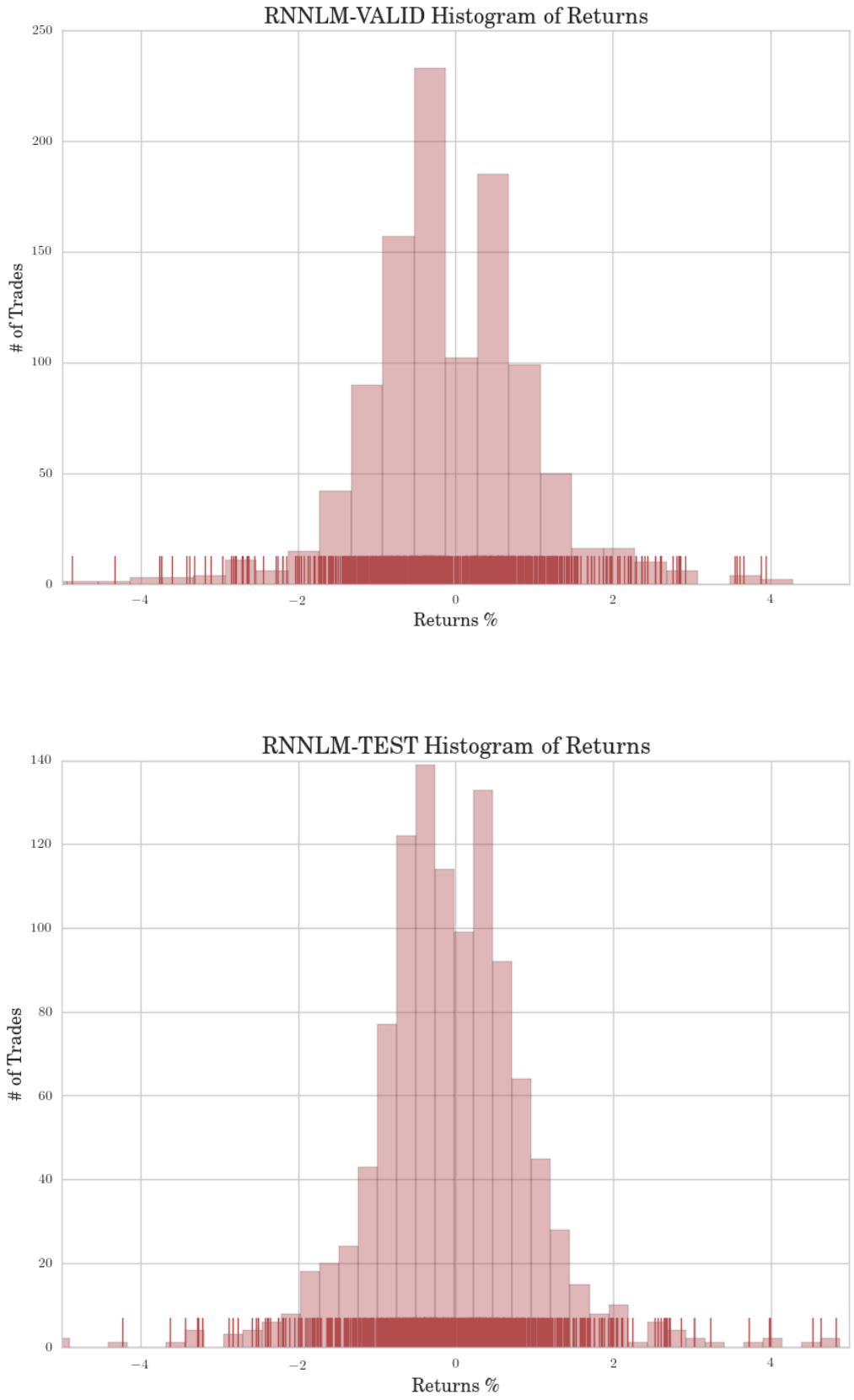


Figure 5.21: RNNLM Histogram of Percentage Returns Focused on Range (-5%,5%)

Table 5.4: 60-Minute Language Model Metrics on Validation Set

Model Type	Profit Per Share ($)	Returns (%)	Risk	Trades	Sharpe Ratio	Precision	Recall	Accuracy
NBSVM	0.761	0.686	1.432	761	7.602	0.612	0.057	0.556
RNNLM	-0.051	-0.125	1.552	1068	-1.282	0.591	0.038	0.544
PARAGRAPH	0.698	0.565	1.784	865	5.025	0.621	0.060	0.557
NBSVM RNNLM	0.970	0.798	0.788	42	16.080	0.582	0.003	0.537
NBSVM PARAGRAPH	0.832	0.717	1.712	389	6.648	0.599	0.029	0.546
RNNLM PARAGRAPH	0.988	0.628	1.013	84	9.838	0.611	0.006	0.538
NBSVM RNNLM PARAGRAPH	0.867	0.698	1.596	479	6.946	0.607	0.036	0.549
RANDOM-AVG	0.001	0.001	1.299	2150	0.011	0.290	0.050	0.505

Table 5.5: 60-Minute Language Model Metrics on Testing Set

Model Type	Profit Per Share ($)	Returns (%)	Risk	Trades	Sharpe Ratio	Precision	Recall	Accuracy
NBSVM	0.752	0.636	0.918	820	10.990	0.558	0.057	0.563
RNNLM	-0.086	-0.037	1.228	1109	-0.481	0.539	0.036	0.549
PARAGRAPH	0.561	0.500	1.125	818	7.058	0.577	0.054	0.562
NBSVM RNNLM	0.824	0.690	1.213	72	9.028	0.477	0.004	0.547
NBSVM PARAGRAPH	0.770	0.654	0.892	430	11.653	0.553	0.030	0.555
RNNLM PARAGRAPH	0.514	0.352	1.986	89	2.815	0.574	0.006	0.548
NBSVM RNNLM PARAGRAPH	0.752	0.602	1.185	521	8.067	0.553	0.036	0.556
RANDOM-AVG	-0.002	0.003	1.013	2237	0.042	0.288	0.050	0.514

Table 5.6: 60-Minute Regression Model Metrics on Validation Set

Model Type	Profit Per Share ($)	Returns (%)	Risk	Trades	Sharpe Ratio	Precision	Recall	Accuracy
LR	0.038	0.330	2.805	442	1.870	0.438	0.060	0.598
SVR	-0.044	0.192	3.585	251	0.852	0.455	0.032	0.593
KRR	0.041	0.261	3.053	349	1.356	0.421	0.046	0.596
FWD 0 RNN 6 Dense	-0.040	0.104	3.028	373	0.548	0.362	0.043	0.591
RNN 1 GRU 1 Dense	0.181	0.077	2.724	482	0.448	0.399	0.059	0.593
RNN 4 LSTM 1 Dense	0.158	0.904	4.431	128	3.237	0.432	0.018	0.591
RNN 7 LSTM 0 Dense	0.120	0.240	4.102	176	0.928	0.413	0.023	0.590
RANDOM-AVG	0.005	-0.002	0.839	768	-0.027	0.226	0.050	0.633

Table 5.7: 60-Minute Regression Model Metrics on Testing Set

Model Type	Profit Per Share ($)	Returns (%)	Risk	Trades	Sharpe Ratio	Precision	Recall	Accuracy
LR	0.032	0.024	1.311	332	0.291	0.432	0.047	0.602
SVR	0.079	0.250	1.555	155	2.553	0.508	0.024	0.601
KRR	0.004	-0.022	1.381	273	-0.249	0.417	0.038	0.602
FWD 0 RNN 6 Dense	-0.014	0.031	1.263	287	0.384	0.419	0.038	0.599
RNN 1 GRU 1 Dense	0.150	0.078	1.235	408	1.002	0.435	0.052	0.599
RNN 4 LSTM 1 Dense	0.176	0.243	1.402	78	2.746	0.447	0.012	0.598
RNN 7 LSTM 0 Dense	-0.313	0.090	1.892	111	0.755	0.241	0.015	0.597
RANDOM-AVG	-0.003	0.000	0.596	740	0.011	0.217	0.050	0.648

region of prices to minute changes also normalizes prices. Stock movements that trade at higher prices still may be elevated, but the differences will have a much smaller scale.

The variance in the price an asset trades at, explains why some strategies have a negative profit per share but positive returns. For example, on the validation set, SVR loses an average of -0.04 cents per trade but has an average of 0.19% returns. SVR made a few bad predictions on stocks that trade at higher prices; this could be a large decline in cents but a small ratio according to percentage return. With only 251 transactions, it is not fair to separate evaluation by price range. However, given a larger dataset, it would be encouraged to avoid disparities by selecting small price ranges. Nonetheless, the current regression techniques learn to be profitable with the selected representation. However, only one regression model, RNN 4 LSTM 1 Dense, achieves a Sharpe Ratio of over 2 across both validation and testing set.

K-NN with DTW is the only time-series model that is not a regression model. K-NN's lengthy execution time lessens its value. Training time is the bottleneck for regression and language models. Alternatively, for K-NN the bottleneck is generating representatives and predicting samples. Due to the bottlenecks, only values of 1 and 3 were selected for K to assess performance on a 60-minute exit strategy. 1-NN achieves a Sharpe Ratio of 0.965 with 1116 trades on the validation set, and a Sharp Ratio of 0.195 with 1020 trades on the testing set. When K is increased to 3, we obtain negative Sharpe Ratios and fewer transactions (Sharpe Ratios of -1.11 and -0.73 for validation and testing). Since the time required to compute is long, we did not use the full training set for checking neighbors. Instead, using the design stated previously, we generated 25 representatives for Buy, 25

for Short, and 250 Neutral. For investors that use technical analysis, this should not be a reason not to try DTW. We are only looking at a subset consisting of noisy 30-minute period intraday events, but smoothing of DTW may hold better for less noisy and more prominent trading patterns.

5.7 Ensembles of Time-Series Models

Ensemble models are investigated since most individual regression techniques do not approach ample Sharpe Ratios. Model combinations and weights are selected using the method described earlier. Listed in Table 5.8 are the top 10 performers based on Sharpe Ratio on the testing set.

As shown in Tables 5.9 and 5.10, all of the recorded ensembles have Sharpe Ratios greater than two. Of the 10, seven use RNN 4 LSTM 1 Dense. Three out of the seven, 1, 2, and 4, perform better as a combination than the original.

All ensembles that do not use RNN 4 LSTM 1 Dense include SVR and RNN 1 GRU 1 Dense. Ensembling improves performance. But given similarities in model combinations, it is logical only to focus on Ensemble 1, Ensemble 2, and Ensemble 3. There are a few downsides for the regression models. First, there are a limited number of trading opportunities in both datasets. Second, while all returns are positive, there is a significant difference in the returns and risk from the validation set to the testing set. The difference may foretell a potential instability to changing financial conditions. An evaluation of a larger number of trades would help alleviate such concerns.

Table 5.8: Top Ensembles

Ensemble 1	FWD 0 RNN 6 Dense & RNN 4 LSTM 1 Dense
Ensemble 2	RNN 4 LSTM 1 Dense & RNN 1 GRU 1 Dense & RNN 7 LSTM 0 Dense
Ensemble 3	RNN 1 GRU 1 Dense & SVR
Ensemble 4	FWD 0 RNN 6 Dense & RNN 4 LSTM 1 Dense & RNN 1 GRU 1 Dense & SVR
Ensemble 5	FWD 0 RNN 6 Dense & RNN 4 LSTM 1 Dense & KRR & RNN 1 GRU 1 Dense & SVR
Ensemble 6	FWD 0 RNN 6 Dense & RNN 4 LSTM 1 Dense & RNN 1 GRU 1 Dense & LR & RNN 7 LSTM 0 Dense & SVR
Ensemble 7	RNN 4 LSTM 1 Dense & RNN 1 GRU 1 Dense
Ensemble 8	FWD 0 RNN 6 Dense & RNN 1 GRU 1 Dense & SVR
Ensemble 9	FWD 0 RNN 6 Dense & RNN 4 LSTM 1 Dense & KRR & SVR
Ensemble 10	FWD 0 RNN 6 Dense & RNN 1 GRU 1 Dense & RNN 7 LSTM 0 Dense & SVR

Table 5.9: 60-Minute Ensemble of Regression Models Metrics on Validation Set

Model Type	Profit Per Share ($)	Returns (%)	Risk	Trades	Sharpe Ratio	Precision	Recall	Accuracy
Ensemble 1	0.064	0.679	4.719	123	2.285	0.410	0.016	0.590
Ensemble 2	0.100	0.693	4.775	119	2.305	0.441	0.016	0.590
Ensemble 3	-0.001	0.591	4.399	145	2.132	0.423	0.019	0.591
Ensemble 4	0.060	0.616	4.771	116	2.051	0.409	0.015	0.591
Ensemble 5	0.226	0.900	4.773	120	2.994	0.460	0.018	0.592
Ensemble 6	0.212	0.992	5.055	104	3.115	0.424	0.014	0.590
Ensemble 7	0.069	0.653	4.273	142	2.425	0.398	0.018	0.589
Ensemble 8	-0.036	0.608	4.613	137	2.092	0.370	0.017	0.591
Ensemble 9	0.147	0.813	4.713	123	2.737	0.436	0.017	0.591
Ensemble 10	0.182	0.857	5.001	106	2.721	0.396	0.016	0.591

Table 5.10: 60-Minute Ensemble of Regression Models Metrics on Testing Set

Model Type	Profit Per Share ($)	Returns (%)	Risk	Trades	Sharpe Ratio	Precision	Recall	Accuracy
Ensemble 1	0.151	0.316	1.449	75	3.464	0.473	0.012	0.599
Ensemble 2	0.047	0.269	1.332	68	3.210	0.468	0.011	0.598
Ensemble 3	0.276	0.327	1.677	105	3.098	0.531	0.019	0.601
Ensemble 4	0.184	0.295	1.546	72	3.031	0.484	0.012	0.599
Ensemble 5	0.077	0.238	1.427	83	2.653	0.509	0.015	0.600
Ensemble 6	0.069	0.248	1.513	68	2.605	0.515	0.013	0.600
Ensemble 7	0.194	0.210	1.372	89	2.427	0.428	0.013	0.598
Ensemble 8	0.112	0.244	1.625	96	2.382	0.539	0.018	0.601
Ensemble 9	0.052	0.218	1.465	80	2.366	0.513	0.014	0.599
Ensemble 10	-0.165	0.238	1.773	66	2.135	0.537	0.013	0.599

5.8 Summary

The event clustering algorithm does well in generating events for both company-centric and energy events. However, learning algorithms only perform well on company-centric events. A high velocity of generated clusters, negatively impacts the learning algorithms' ability to identify the event that influences the stock price over events that do not. Because of this confusion, energy events are difficult to predict. Consequently, only predictions from company-centric events are used in trading.

Both language and regression models have positive Sharp Ratios. Language models are better predictors and more profitable than regression models. Language models have a larger training set available for learning since they do not require financial data for predictions. Performance on regression models is strengthened by taking advantage of ensembles.

Table 5.11 contains a list of recommended models along with their reported Sharpe Ratios. Recommendations reflect evaluation metrics and the distribution of returns. Lastly, language models are favored. The preference is not only due to greater Sharpe Ratios but also because performance is reported over a larger sample of trading decisions. Reporting performance on many trades lifts belief in the models ability to stay persistent.

Table 5.11: Recommended Models

Model	Type	Sharpe Valid	Sharpe Test
NBSVM	Language	7.602	10.990
PARAGRAPH	Language	5.025	7.058
NBSVM PARAGRAPH	Language	6.648	11.653
NBSVM RNNLM PARAGRAPH	Language	6.946	8.067
FWD 0 RNN 6 Dense & RNN 4 LSTM 1 Dense	Regression	2.285	3.464
RNN 4 LSTM 1 Dense & RNN 1 GRU 1 Dense & RNN 7 LSTM 0 Dense	Regression	2.305	3.210
RNN 1 GRU 1 Dense & SVR	Regression	2.132	3.098

5.9 Statistical Analysis of Recommended Models

The distribution of returns for recommended models confirms profitability. Statistical significance tests are conducted in order to determine if the distributions are statistically different. In comparing multiple models, the Friedman test is used to check if there is a difference in returns amongst all models on repeated predictions [Friedman 1939]. If the hypothesis shows that returns are statistically different, then we use the Mann-Whitney-Wilcoxon test to do a post-hoc analysis between all pairs of models [Mann & Whitney 1947]. The selection of the two tests is conservative because neither requires a normality assumption. Still, there are limitations as we are only using one dataset and cross-validation may not be appropriate for financial time-series. As language and time-series models' trade on different subsets, they are looked at separately. All tests are measured on both validation and testing sets.

Friedman tests on the three time-series ensembles return a value of $P > 0.05$, so there is no need to complete a post-hoc analysis on each pair. We can not conclude that the distribution differs for regression ensembles. An explanation may be because time-series combinations use weighted averages, which share significant influences from particular

Table 5.12: P-values from Mann-Whitney-Wilcoxon Test on Language Model Pairs

Model 1	Model 2	Valid Set p-value	Test Set p-value
NBSVM	PARAGRAPH	0.645	0.556
NBSVM	NBSVM & PARAGRAPH	0.025	0.039
NBSVM	NBSVM & RNNLM & PARAGRAPH	0.082	0.096
PARAGRAPH	NBSVM & PARAGRAPH	0.077	0.141
PARAGRAPH	NBSVM & RNNLM & PARAGRAPH	0.204	0.285
NBSVM & PARAGRAPH	NBSVM & RNNLM & PARAGRAPH	0.615	0.688

models.

Running the Friedman test on the four language models returns values of $P < 0.01$ on both validation and testing set. For this reason, the p-values from the Mann-Whitney-Wilcoxon test is computed and listed in Table 5.12. Of the six pairs, only one of them is statistically different when using a 0.05 critical p-value. While both NBSVM and PARAGRAPH are profitable individually, we can not conclude that their difference is statistically significant in terms of return distributions. However, the two statistically differ from the distributions generated by the RNNLM model. Statistical tests and box plots of returns can aid in verifying model recommendations.

Chapter 6
Conclusions

6.1 Discussion of Hypotheses

Hypothesis 1: *In order to move beyond general group sentiment on Twitter, one needs to determine how to use tweets similar to news headlines. Thus, for tweets, keyword-based similarity clusters will generate Twitter events that resemble headlines.*

Journalists scour through dealings and stories to find noteworthy news headlines. Aside from a character limit, individual tweets have no restrictions – thereby providing a lot of useless information to sift through for noteworthy items. Those using Twitter as a source of news for trading, have to either filter through every tweet or use a general sentiment from all tweets. The implemented event clustering algorithm sequentially assembles tweets with similar text; creating topical events as clusters. Two collections use the technique, one set for the energy sector, and one that mentions corporations using the Twitter "$" symbol. Events associate a timestamp as well as a stock symbol in its metadata if specified. Tradeable timestamps are used to evaluate the influence of events to stock prices. Clusters do not guarantee removal of useless items since Twitter is full of casual conversation. However, only gathering related tweets under specific time restrictions help identify developing stories. Furthermore, filters reduce junk by removing retweets and tweets from some automated bots. A keyword-based similarity clustering technique used on a total of 87,468,657 tweets forms 116,919 events. As

such, the clustering technique is useful in generating Twitter events that resemble news headlines. Therefore, this hypothesis is confirmed.

Hypothesis 2: *Given Twitter events, it is possible to address Pre-News and Lagged-News concerns.*

A pre-news effect, where prices move before the release of news headlines, occurs in many studies that use daily frequencies. Our learning algorithms trade on intraday frequencies of minutes on Twitter generated events. The majority of generated events have an associated timestamp of 9:30 am EST when financial markets open. Therefore, prices have not had a chance to incorporate any information found in events. By making decisions before trading hours, we avoid pre-news concerns. Any trades not at market open may or may not have a pre-news effect; accounting for intraday noise in movements is hard.

To address lagged-news, we have used an approach of merging events. We purge all matching clusters that arrive within a 4-hour time-frame of a saved event. For instance, in this study, there are 201,631 cases merged, which is important considering a total of only 116,919 events created. A key benefit of merging is to avoid repeated purchases on a single story. Multiple transactions would have caused serial correlations in returns, which were previously shown to overstate Sharpe Ratios. Furthermore, if the merged event arrives a few hours late, the price influence may have already been exhausted. For language models, merging removes tweets with texts that are similar but have different labels, and thus makes the learning task simpler. The hypothesis is confirmed due to

mechanisms in place to address pre-news and lagged-news. However, no measure of how much the effects dissipate is presented.

Hypothesis 3: *When using modern methods on text and prices (in their original representation), the performance of a trading strategy – on Twitter generated events – that predicts stock price jumps can be profitable.*

Methods trained are successful on the company-centric collection but do not produce beneficial results for general events in the energy sector. Therefore, seven models have been recommended for the company-centric collection. Recommendations are listed in Table 5.11.

The suggested models are profitable; four are text-based, and three use time-series. All learning algorithms evaluate performance on two months of unseen data, the validation set and testing set. Sharpe Ratios are used to gauge profitability. By general standards, Sharpe Ratios greater than 3 are considered excellent. Advised models have Sharpe Ratios ranging from 2.1 to 7.6 for the validation set, and 3.1 to 11.7 on the testing set. Text models are preferred over price-based models. Of the language models, PARAGRAPH uses a distributed representation of words; this type of model has not been studied in Event Studies. Regression models keep 30-minutes of pre-event region data in its entirety to predict future movements. The representation used for time-series is a minute by minute price shift; returns and log returns were found to be too noisy for prediction. All three recommended regression models are ensembles that contain at least one Recurrent Neural Network architecture. The energy collection does not provide profitable results.

Due to profitability, the hypothesis is confirmed for Twitter generated events for the company-centric collection.

6.2 Contributions

Given the evolution of news mediums and how information disseminates to the public, our research aids Event Studies in Finance by adopting social content in a manner similar to that of traditional news headlines. Current techniques for social content in trading use general sentiment translated as numeric time-series. We present an event clustering algorithm that uses keyword-based similarity clusters to identify developing events. A method to gather domain expertise for the energy sector is presented. Recommendations for filtering tweets that hurt event generation are given.

We design routines to connect events with tradeable securities in a way that allows intraday trading. Text and price-based models that are considered the state-of-the-art are implemented to create prediction systems for trading decisions. Evaluation metrics that bridge a gap between industry standards and academia are presented. All design choices are structured and defined in a way that helps create useful trading systems. For example, metrics are not concerned with general ability to predict direction but instead classify valuable price jumps. Empirical results between various intraday exit strategies are presented.

In summary, the research generates events from social content, predicts trading decisions on those events, and evaluates learning algorithms to produce recommendations for profitable trading strategies.

6.3 Limitations and Future Work

A common limitation for financial problems is the time-span of collected data. Changing market conditions always cast doubt on any models' ability to stay persistent. Choosing intraday periods aid in increasing the volume of our collections. However, the desired trade count of around 100 may not seem sufficient for regression models. Still, the small number of trades have to be taken alongside the context that prediction was solely based on Twitter generated events.

Recurrent Neural Network architectures need further testing in future work of time-series predictions in financial markets. The architecture has shown promise in this work. It will be beneficial to study varying neural network designs in short-term market predictions on large datasets.

The size of the dataset is another limitation. While the numbers of tweets used and events collected were sufficient, more is better. Given a larger dataset, we could separate all events by price range. For example, stocks that trade between $90 to $100 may not need normalization and have easier patterns for learning algorithms. Smaller price ranges would make automatic labeling of events simpler for supervised learning. Also, deep learning language models like RNNLM are known to improve with larger amounts of data. In this work RNNLM performed poorly, this may not be the case given more training data.

The event clustering algorithm takes days to run on over 80 million tweets. Due to this constraint, parameters for the event clustering algorithm have not been tuned. Variables include cluster size, cluster time, merge time and matching criteria. Currently,

event clustering uses a similarity matching rule of exact matching of Part of Speech tags. It will be interesting to leverage cosine similarity on distributed representation of word embeddings to find close word matches numerically. Word embeddings can be trained on all tweets using the word2vec tool [Mikolov *et al.* 2013a]. Any improvements to the event clustering algorithm should benefit machine learning predictions.

New mediums of delivering information to the public continue to emerge; to adapt, Event Analysis and Computational Finance need to promote the generation of events from unstructured data. Most questions answered previously looked at structured data for events; such as earnings announcements. Now techniques to identify entities and cluster related events from casual conversation can move the research forward to readjust to current and future social media content. Furthermore, machine formed actionable trading decisions may improve as natural language processing and time-series classification techniques advance.

Appendix A

Appendix

A.1 Appendix

Table A.1: Energy Symbols Tracked by Twitter

ALJ	AM	APA	APC	AR	ARLP	ATW	BHI	BP
BPL	BRGYY	BRS	BSM	BTE	BWP	CAM	CCJ	CHK
CIE	CJES	CLB	CLMT	CLR	CMLP	CNQ	CNX	COG
COP	CPG	CPLP	CPPL	CRC	CRZO	CVE	CVI	CVRR
CVX	CXO	CZZ	DK	DNR	DO	DPM	DRQ	DVN
E	EC	ECA	EEP	EGN	ENB	ENLK	EOG	EPD
EPE	EQT	ERF	ESV	ETE	ETP	EURN	EXH	FANG
FET	FI	FTI	GEL	GLNG	GLOG	GPOR	HAL	HES
HFC	HP	IMO	INT	IOC	KMI	KOS	LINE	LNG
LPI	MDR	MMP	MPC	MPLX	MRD	MRO	MTDR	MUR
MWE	NAT	NBL	NBR	NE	NFX	NGL	NGLS	NOV
NS	NTI	OAS	OII	OIS	OKE	OKS	OXY	PAA
PAGP	PBA	PBF	PBR	PBR/A	PDCE	PDS	PE	PSX
PTEN	PXD	QEP	RDC	RDS/A	RDS/B	RES	RICE	RIG
RRC	RSPP	SDLP	SDRL	SE	SEMG	SEP	SFL	SHLX
SLB	SLCA	SM	SMLP	SPN	SSL	STNG	STO	SU
SUN	SWN	SXL	SYRG	TK	TLLP	TOO	TOT	TRGP
TRP	TS	TSO	UGP	UNT	UPL	VLO	VNR	WES
WFT	WLL	WMB	WNR	WPX	WPZ	XEC	XOM	YPF

Table A.2: S&P 500 Symbols Tracked by Twitter

A	AAL	AAPL	ABBV	ABC	ABT	ACE	ACN	ADBE
ADI	ADM	ADP	ADS	ADSK	ADT	AET	AFL	AGN
AIG	AIV	AIZ	AKAM	ALL	ALLE	ALTR	ALXN	AMAT
AME	AMG	AMGN	AMP	AMT	AMZN	AN	ANTM	AON
APH	AVB	AVGO	AXP	AZO	BA	BAC	BAX	BBBY
BBT	BBY	BCR	BDX	BEN	BF B	BIIB	BK	BLK
BMY	BRCM	BRK B	BSX	BWA	BXP	C	CA	CAG
CAH	CAT	CB	CBG	CBS	CCE	CCI	CCL	CELG
CERN	CHRW	CI	CINF	CL	CLX	CMA	CMCSA	CME
CMG	CMI	COF	COH	COL	COST	CPB	CSC	CSCO
CSX	CTAS	CTL	CTSH	CTXS	CVC	CVS	DAL	DE
DFS	DG	DGX	DHI	DHR	DIS	DISCA	DLPH	DLTR
DNB	DOV	DPS	DRI	DTV	DVA	EA	EBAY	EFX
EL	EMC	EMR	ENDP	EQIX	EQR	ESRX	ESS	ETFC
ETN	EW	EXPD	EXPE	F	FAST	FB	FDO	FDX
FFIV	FIS	FISV	FITB	FLIR	FLR	FLS	FOSL	FOXA
FSLR	FTR	GD	GE	GGP	GILD	GIS	GLW	GM
GMCR	GME	GNW	GOOGL	GPC	GPS	GRMN	GS	GT
GWW	HAR	HAS	HBAN	HBI	HCA	HCBK	HCN	HCP
HD	HIG	HOG	HON	HOT	HPQ	HRB	HRL	HRS
HSIC	HSP	HST	HSY	HUM	IBM	ICE	INTC	INTU
IPG	IR	IRM	ISRG	ITW	IVZ	JCI	JEC	JNJ
JNPR	JOY	JPM	JWN	K	KEY	KIM	KLAC	KMB
KMX	KO	KORS	KR	KRFT	KSS	KSU	L	LB
LEG	LEN	LH	LLL	LLTC	LLY	LM	LMT	LNC
LOW	LRCX	LUK	LUV	LVLT	M	MA	MAC	MAR
MAS	MAT	MCD	MCHP	MCK	MCO	MDLZ	MDT	MET
MHFI	MHK	MJN	MKC	MMC	MMM	MNK	MNST	MO
MRK	MS	MSFT	MSI	MTB	MU	MYL	NAVI	NDAQ
NFLX	NKE	NOC	NSC	NTAP	NTRS	NVDA	NWL	NWSA
O	OMC	ORCL	ORLY	PAYX	PBCT	PBI	PCAR	PCL
PCLN	PCP	PDCO	PEP	PFE	PFG	PG	PGR	PH
PHM	PKI	PLD	PM	PNC	PNR	PRGO	PRU	PSA
PVH	PWR	QCOM	QRVO	R	RAI	RCL	REGN	RF
RHI	RHT	RL	ROK	ROP	ROST	RSG	RTN	SBUX
SCHW	SJM	SLG	SNA	SNDK	SNI	SPG	SPLS	SRCL
STI	STJ	STT	STX	STZ	SWK	SWKS	SYK	SYMC
SYY	T	TAP	TDC	TEL	TGNA	TGT	THC	TIF
TJX	TMK	TMO	TRIP	TROW	TRV	TSCO	TSN	TSS
TWC	TWX	TXN	TXT	TYC	UA	UHS	UNH	UNM
UNP	UPS	URBN	URI	USB	UTX	V	VAR	VFC
VIAB	VNO	VRSN	VRTX	VTR	VZ	WAT	WBA	WDC
WFC	WFM	WHR	WM	WMT	WU	WY	WYN	WYNN
XL	XLNX	XRAY	XRX	XYL	YHOO	YUM	ZBH	ZTS

Table A.3: Blacklisted Sources

<a href="http://ifttt.com" rel="nofollow">
<a href="http://dlvr.it" rel="nofollow">
<a href="http://sandaysoft.com/" rel="nofollow">
<a href="http://www.hootsuite.com" rel="nofollow">
<a href="http://www.bandaigames.channel.or.jp/list/one_main/tc/en/" rel="nofollow">
<a href="http://www.weather-display.com" rel="nofollow">
<a href="http://www.buoyalarm.com" rel="nofollow">
<a href="http://twittbot.net/" rel="nofollow">
<a href="http://twitter.com/USACities/cities" rel="nofollow">
<a href="http://twitter.com/WorldCities/cities" rel="nofollow">
<a href="https://github.com/jim-easterbrook/pywws" rel="nofollow">
<a href="http://www.meteobridge.com" rel="nofollow">
<a href="http://www.weatherstats.ca" rel="nofollow">
<a href="http://www.weatheronline.co.uk/Japan/Tokyo.htm" rel="nofollow">
<a href="http://mesonet.agron.iastate.edu/projects/iembot/" rel="nofollow">
<a href="http://www.ajaymatharu.com/" rel="nofollow">
<a href="http://neuvoo.ca" rel="nofollow">
<a href="http://service.rss2twi.com/" rel="nofollow">
<a href="http://ijg.me" rel="nofollow">
<a href="http://www.forcetweet.nl" rel="nofollow">
<a href="http://www.floodgap.com/software/ttytter/" rel="nofollow">
<a href="http://saratoga-weather.org/scripts-TweetWX.php#TweetWX" rel="nofollow">
<a href="http://share.radionomy.com" rel="nofollow">
<a href="http://www.tsepa.net" rel="nofollow">
<a href="http://tweet2home.com" rel="nofollow">
<a href="http://www.tweetjukebox.com" rel="nofollow">
<a href="https://www.socialoomph.com" rel="nofollow">
<a href="http://www.realtime1.com" rel="nofollow">
<a href="https://twitter.com/iotnet" rel="nofollow">
<a href="http://giveawaytools.com" rel="nofollow">
<a href="http://bufferapp.com" rel="nofollow">
<a href="http://soundhound.com/" rel="nofollow">
<a href="http://spotify.com" rel="nofollow">
<a href="http://gadgetter-se.seesaa.net/" rel="nofollow">
<a href="http://weathercloud.net" rel="nofollow">
<a href="http://foursquare.com" rel="nofollow">
<a href="https://path.com/" rel="nofollow">
<a href="http://129.252.139.134" rel="nofollow">
<a href="http://arduino-tweet.appspot.com/" rel="nofollow">
<a href="http://www.simpleweatheralert.com" rel="nofollow">

Table A.4: Top 20 Sources for Energy Events

Source	Tweet Count
<a href="http://twitterfeed.com" rel="nofollow">	105527
<a href="http://twitter.com" rel="nofollow"	66193
<a href="http://www.facebook.com/twitter" rel="nofollow"	21303
<a href="http://twitter.com/download/iphone" rel="nofollow"	21017
<a href="http://twitter.com/download/android" rel="nofollow"	17842
<a href="http://www.google.com/" rel="nofollow"	16212
<a href="http://publicize.wp.com/" rel="nofollow"	15728
<a href="https://about.twitter.com/products/tweetdeck" rel="nofollow"	12099
<a href="http://www.tweet-eye.com" rel="nofollow"	6623
<a href="http://mobile.twitter.com" rel="nofollow"	4781
<a href="http://twitter.com/#!/download/ipad" rel="nofollow"	4513
<a href="http://linkis.com" rel="nofollow"	4131
<a href="https://mobile.twitter.com" rel="nofollow"	3391
<a href="http://www.apple.com" rel="nofollow"	2149
<a href="http://www.linkedin.com/" rel="nofollow"	2019
<a href="http://twibble.io" rel="nofollow"	1853
<a href="https://mobile.twitter.com" rel="nofollow"	1296
<a href="http://instagram.com" rel="nofollow"	1225
<a href="http://www.jrustonapps.com/apps/my-earthquake-alerts" rel="nofollow"	1224
<a href="http://www.networkedblogs.com/" rel="nofollow"	1045

Table A.5: Top 20 Sources for Company-Centric Events

Source	Tweet Count
<a href="http://twitter.com" rel="nofollow">	23161
<a href="http://seekingalpha.com" rel="nofollow"	8937
<a href="http://twitter.com/download/iphone" rel="nofollow"	5588
<a href="https://about.twitter.com/products/tweetdeck" rel="nofollow"	5267
<a href="http://stocktwits.com" rel="nofollow"	4744
<a href="http://twitterfeed.com" rel="nofollow"	4151
<a href="https://smqueue.com" rel="nofollow"	3974
<a href="http://www.google.com/" rel="nofollow"	2998
<a href="http://investorshangout.com/" rel="nofollow"	2498
<a href="http://127.0.0.1:3000/" rel="nofollow"	2144
<a href="http://www.buzzjust.in/" rel="nofollow"	2064
<a href="http://www.firsttomarkets.com/" rel="nofollow"	1877
<a href="http://twitter.com/download/android" rel="nofollow"	1584
<a href="http://www.advfn.com" rel="nofollow"	1515
<a href="http://investwithconf.blogspot.com" rel="nofollow"	1453
<a href="http://www.wlst.com" rel="nofollow"	1134
<a href="http://www.stocknewswires.com" rel="nofollow"	1021
<a href="http://pocketinfopush.com" rel="nofollow"	982
<a href="http://www.estimize.com" rel="nofollow"	975
<a href="http://twitter.com/#!/download/ipad" rel="nofollow"	913

Table A.6: Top 20 Users for Energy Events

User ID	Tweet Count
3170121190	2148
2373813025	2064
23059499	1983
926793032	1877
207441033	1478
2344623210	1453
15296897	1271
3044159284	1134
2181373892	1105
2825810342	1026
2356524462	982
601422914	975
2243075472	909
61661638	894
102325185	880
528770977	750
565538332	748
201296379	723
4811917159	720
2997788563	670

Table A.7: Top 20 Users for Company-Centric Events

User ID	Tweet Count
1408543818	1477
3137517334	1224
1490345010	1212
167887733	976
107594885	886
2231028101	840
4016028339	832
485528822	824
1414684496	701
3302829610	625
543384934	576
369760961	518
3506094623	505
20993979	489
247171068	488
5950272	488
3457241715	480
248379139	474
2838117555	473
245490052	470

Bibliography

[Antweiler & Frank 2004] Werner Antweiler and Murray Z Frank. *Is all that talk just noise? The information content of internet stock message boards*. The Journal of Finance, vol. 59, no. 3, pages 1259–1294, 2004. (Cited on page 10.)

[Berndt & Clifford 1994] Donald J Berndt and James Clifford. *Using Dynamic Time Warping to Find Patterns in Time Series.* In Knowledge Discovery and Data Mining Workshop, volume 10, pages 359–370. ACM, 1994. (Cited on page 27.)

[Bollen *et al.* 2011] Johan Bollen, Huina Mao and Xiaojun Zeng. *Twitter mood predicts the stock market.* Journal of Computational Finance, vol. 2, pages 1–8, 2011. (Cited on pages 8, 11 and 20.)

[Chan 2003] Wesley S Chan. *Stock price reaction to news and no-news: drift and reversal after headlines.* Journal of Financial Economics, vol. 70, no. 2, pages 223–260, 2003. (Cited on pages 8 and 9.)

[Chan 2008] Ernie Chan. Quantitative trading: how to build your own algorithmic trading business. John Wiley & Sons, 2008. (Cited on page 43.)

[Chatrath *et al.* 2014] Arjun Chatrath, Hong Miao, Sanjay Ramchander and Sriram Villupuram. *Currency jumps, cojumps and the role of macro news.* Journal of International Money and Finance, vol. 40, pages 42–62, 2014. (Cited on page 10.)

[Chung *et al.* 2012] San-Lin Chung, Chi-Hsiou Hung and Chung-Ying Yeh. *When does investor sentiment predict stock returns?* Journal of Empirical Finance, vol. 19, no. 2, pages 217–240, 2012. (Cited on page 10.)

[Daniel *et al.* 1998] Kent Daniel, David Hirshleifer and Avanidhar Subrahmanyam. *Investor psychology and security market under-and overreactions.* The Journal of Finance, vol. 53, no. 6, pages 1839–1885, 1998. (Cited on page 5.)

[Davis & Goadrich 2006] Jesse Davis and Mark Goadrich. *The relationship between Precision-Recall and ROC curves.* In Proceedings of the 23rd International Conference on Machine Learning, pages 233–240. ACM, 2006. (Cited on page 39.)

[Fama 1965] Eugene F Fama. *The behavior of stock-market prices.* Journal of Business, vol. 38, pages 34–105, 1965. (Cited on page 4.)

[Fasanghari & Montazer 2010] Mehdi Fasanghari and Gholam Ali Montazer. *Design and implementation of fuzzy expert system for Tehran Stock Exchange portfolio recommendation.* Expert Systems with Applications, vol. 37, no. 9, pages 6138–6147, 2010. (Cited on page 10.)

[Fidelity 2016] Fidelity. `http://research2.fidelity.com/fidelity/screeners/commonstock/main.asp?`, 2016. (Cited on page 35.)

[Friedman 1939] Milton Friedman. *A Correction: The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance.* Journal of the

American Statistical Association, vol. 34, no. 205, page 109, 1939. (Cited on page 93.)

[Ghiassi *et al.* 2013] M Ghiassi, J Skinner and D Zimbra. *Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network.* Expert Systems with Applications, vol. 40, no. 16, pages 6266–6282, 2013. (Cited on page 10.)

[Giusti & Batista 2013] Roberto Giusti and Gustavo E Batista. *An empirical comparison of dissimilarity measures for time series classification.* In Intelligent Systems, pages 82–88. IEEE, 2013. (Cited on pages 24 and 27.)

[Goodfellow *et al.* 2016] Ian Goodfellow, Yoshua Bengio and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016. (Cited on page 59.)

[GroB-KluBmann & Hautsch 2011] Axel GroB-KluBmann and Nikolaus Hautsch. *When machines read the news: Using automated text analytics to quantify high frequency news-implied market reactions.* Journal of Empirical Finance, vol. 18, no. 2, pages 321–340, 2011. (Cited on pages 7 and 8.)

[Groth & Muntermann 2011] Sven S Groth and Jan Muntermann. *An intraday market risk management approach based on textual analysis.* Decision Support Systems, vol. 50, no. 4, pages 680–691, 2011. (Cited on page 10.)

[Hagenau *et al.* 2013] Michael Hagenau, Michael Liebmann and Dirk Neumann. *Automated news reading: Stock price prediction based on financial news using*

context-capturing features. Decision Support Systems, vol. 55, no. 3, pages 685–697, 2013. (Cited on page 10.)

[Huang *et al.* 2010] Chenn-Jung Huang, Jia-Jian Liao, Dian-Xiu Yang, Tun-Yu Chang and Yun-Cheng Luo. *Realization of a news dissemination agent based on weighted association rules and text mining techniques*. Expert Systems with Applications, vol. 37, no. 9, pages 6409–6413, 2010. (Cited on page 10.)

[Kontopoulos *et al.* 2013] Efstratios Kontopoulos, Christos Berberidis, Theologos Dergiades and Nick Bassiliades. *Ontology-based sentiment analysis of twitter posts*. Expert Systems with Applications, vol. 40, no. 10, pages 4065–4074, 2013. (Cited on page 10.)

[Le & Mikolov 2014] Quoc V Le and Tomas Mikolov. *Distributed Representations of Sentences and Documents*. In ICML, pages 1188–1196. JMLR, 2014. (Cited on page 56.)

[Leinweber & Sisk 2011] David Leinweber and Jacob Sisk. *Event Driven Trading and the 'New News'*. Journal of Portfolio Management, vol. 38, no. 1, pages 110–124, 2011. (Cited on pages 7, 8, 10 and 11.)

[Levy *et al.* 2014] Omer Levy, Yoav Goldberg and Israel Ramat-Gan. *Linguistic regularities in sparse and explicit word representations*. Conference on Natural Language Learning, page 171, 2014. (Cited on page 19.)

[Liao *et al.* 2014] Wenhui Liao, Sameena Shah and Masoud Makrehchi. *Winning by following the winners: Mining the behaviour of stock market experts in social media.* In International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction, pages 103–110. Springer, 2014. (Cited on page 10.)

[Lo 2002] Andrew W Lo. *The Statistics of Sharpe Ratios.* Financial Analysts Journal, vol. 58, no. 4, pages 36–52, 2002. (Cited on page 72.)

[MacKinlay 1997] A Craig MacKinlay. *Event Studies in Economics and Finance.* Journal of Economic Literature, vol. 35, no. 1, pages pp. 13–39, 1997. (Cited on pages 1, 4 and 7.)

[Makrehchi *et al.* 2013] Masoud Makrehchi, Sameena Shah and Wenhui Liao. *Stock prediction using event-based sentiment analysis.* In Web Intelligence and Intelligent Agent Technologies, pages 337–342. IEEE, 2013. (Cited on pages 8, 11 and 12.)

[Malkiel 2003] Burton G Malkiel. *The efficient market hypothesis and its critics.* Journal of Economic Perspectives, vol. 17, no. 1, pages 59–82, 2003. (Cited on page 5.)

[Mann & Whitney 1947] H B Mann and D R Whitney. *On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other.* The Annals of Mathematical Statistics, vol. 18, no. 1, pages 50–60, 1947. (Cited on page 93.)

[Mao *et al.* 2013] Yuexin Mao, Wei Wei and Bing Wang. *Twitter volume spikes: analysis and application in stock trading.* In Proceedings of the 7th Workshop on Social Network Mining and Analysis, page 4. ACM, 2013. (Cited on page 10.)

[Mesnil *et al.* 2014] Grégoire Mesnil, Tomas Mikolov, Marc'Aurelio Ranzato and Yoshua Bengio. *Ensemble of Generative and Discriminative Techniques for Sentiment Analysis of Movie Reviews*. Computing Research Repository, vol. abs/1412.5335, 2014. (Cited on pages 44 and 56.)

[Mikolov *et al.* 2010] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernock\'y and Sanjeev Khudanpur. *Recurrent neural network based language model.* In Interspeech, pages 1045–1048. Springer, 2010. (Cited on page 56.)

[Mikolov *et al.* 2013a] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado and Jeff Dean. *Distributed representations of words and phrases and their compositionality.* In Advances in Neural Information Processing Systems, pages 3111–3119. MIT Press, 2013. (Cited on pages 16, 17 and 100.)

[Mikolov *et al.* 2013b] Tomas Mikolov, Wen-tau Yih and Geoffrey Zweig. *Linguistic Regularities in Continuous Space Word Representations*. In HLT-NAACL, pages 746–751. ACM, 2013. (Cited on page 17.)

[Mittermayer 2004] M-A Mittermayer. *Forecasting intraday stock price trends with text mining techniques*. In In System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference, pages 10—-pp. IEEE, 2004. (Cited on page 10.)

[MongoDB 2016] MongoDB. `https://www.mongodb.org`, 2016. (Cited on page 46.)

[Nasseri *et al.* 2015] Alya Al Nasseri, Allan Tucker and Sergio de Cesare. *Quantifying StockTwits semantic terms' trading behavior in financial markets: An effective*

application of decision tree algorithms. Expert Systems with Applications, vol. 42, no. 23, pages 9192–9210, 2015. (Not cited.)

[Nassirtoussi *et al.* 2014] Arman Khadjeh Nassirtoussi, Saeed Aghabozorgi, Teh Ying Wah and David Chek Ling Ngo. *Text mining for market prediction: A systematic review*. Expert Systems with Applications, vol. 41, no. 16, pages 7653–7670, 2014. (Cited on page 14.)

[Nuij *et al.* 2014] Wijnand Nuij, Viorel Milea, Frederik Hogenboom, Flavius Frasincar and Uzay Kaymak. *An automated framework for incorporating news into stock trading strategies*. IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 4, pages 823–835, 2014. (Cited on page 10.)

[Owoputi *et al.* 2013] Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider and Noah A Smith. *Improved part-of-speech tagging for online conversational text with word clusters*. ACM, 2013. (Cited on page 51.)

[Pedregosa *et al.* 2011] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, vol. 12, pages 2825–2830, 2011. (Cited on page 59.)

[QuantQuote 2016] QuantQuote. `https://quantquote.com`, 2016. (Cited on page 46.)

[Rachlin *et al.* 2007] Gil Rachlin, Mark Last, Dima Alberg and Abraham Kandel. *ADMIRAL: A data mining based financial trading system.* In Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on, pages 720–725. IEEE, 2007. (Cited on page 8.)

[Rakthanmanon *et al.* 2012] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria and Eamonn Keogh. *Searching and mining trillions of time series subsequences under dynamic time warping.* In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 262–270. ACM, 2012. (Cited on page 28.)

[Ruiz *et al.* 2012] Eduardo J Ruiz, Vagelis Hristidis, Carlos Castillo, Aristides Gionis and Alejandro Jaimes. *Correlating financial time series with micro-blogging activity.* In Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, pages 513–522. ACM, 2012. (Cited on pages 8 and 10.)

[Schumaker & Chen 2009] Robert P Schumaker and Hsinchun Chen. *Textual analysis of stock market prediction using breaking financial news: The AZFin text system.* ACM Transactions on Information Systems, vol. 27, no. 2, page 12, 2009. (Cited on pages 8, 20 and 21.)

[Schumaker *et al.* 2012] Robert P Schumaker, Yulei Zhang, Chun-Neng Huang and Hsinchun Chen. *Evaluating sentiment in financial news articles.* Decision Support Systems, vol. 53, no. 3, pages 458–464, 2012. (Cited on page 10.)

[Sharpe 1994] William F Sharpe. *The sharpe ratio*. The Journal of Portfolio Management, vol. 21, no. 1, pages 49–58, 1994. (Cited on page 42.)

[Soni *et al.* 2007] Ankit Soni, Nees Jan van Eck and Uzay Kaymak. *Prediction of stock price movements based on concept map information*. In Computational Intelligence in Multicriteria Decision Making, IEEE Symposium on, pages 205–211. IEEE, 2007. (Cited on page 8.)

[Sul *et al.* 2014] Hongkee Sul, Alan R Dennis and Lingyao Ivy Yuan. *Trading on Twitter: The Financial Information Content of Emotion in Social Media*. In System Sciences (HICSS), 2014 47th Hawaii International Conference on, pages 806–815. IEEE, 2014. (Cited on pages 8 and 12.)

[Tetlock *et al.* 2008] Paul C Tetlock, MAYTAL SAAR-TSECHANSKY and Sofus Macskassy. *More than words: Quantifying language to measure firms' fundamentals*. The Journal of Finance, vol. 63, no. 3, pages 1437–1467, 2008. (Cited on pages 7 and 8.)

[Tetlock 2007] Paul C Tetlock. *Giving content to investor sentiment: The role of media in the stock market*. The Journal of Finance, vol. 62, no. 3, pages 1139–1168, 2007. (Cited on page 10.)

[Tsay 2005] Ruey S Tsay. Analysis of financial time series, volume 543. John Wiley & Sons, 2005. (Cited on pages 21, 22 and 23.)

[Twitter 2016] Twitter. `https://twitter.com`, 2016. (Cited on page 1.)

[UCR-Suite 2016] UCR-Suite. http://www.cs.ucr.edu/~eamonn/UCRsuite.html/, 2016. (Cited on page 28.)

[Vanstone & Finnie 2010] Bruce Vanstone and Gavin Finnie. *Enhancing stockmarket trading performance with ANNs*. Expert Systems with Applications, vol. 37, no. 9, pages 6602–6610, 2010. (Cited on page 10.)

[Wang & Manning 2012] Sida Wang and Christopher D Manning. *Baselines and bigrams: Simple, good sentiment and topic classification*. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, pages 90–94. ACM, 2012. (Cited on pages 15, 16 and 56.)

[Word2Vec 2016] Word2Vec. https://code.google.com/p/word2vec/, 2016. (Cited on page 17.)

[Yu *et al.* 2013] Yang Yu, Wenjing Duan and Qing Cao. *The impact of social and conventional media on firm equity value: A sentiment analysis approach*. Decision Support Systems, vol. 55, no. 4, pages 919–926, nov 2013. (Cited on page 10.)

[Zhai *et al.* 2007] Yuzheng Zhai, Arthur Hsu and Saman K Halgamuge. *Combining news and technical indicators in daily stock price trends prediction*. In Advances in Neural Networks–ISNN 2007, pages 1087–1096. Springer, 2007. (Cited on page 20.)